AFRL-IF-RS-TR-2002-62
Final Technical Report
April 2002

THE JFACC PLANNER/SCHEDULER

Carnegie Mellon University and SRI International

**Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. F061 & F231**

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2002-62 has been reviewed and is approved for publication.

APPROVED: John Spina

JOHN SPINA
Project Engineer

FOR THE DIRECTOR: Michael Talbert

MICHAEL L. TALBERT, Maj, USAF
Technical Advisor, Information Technology Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** Apr 02	**3. REPORT TYPE AND DATES COVERED** Final Apr 97 – Dec 01

4. TITLE AND SUBTITLE

THE JFACC PLANNER/SCHEDULER

5. FUNDING NUMBERS

C - F30602-97-C-0066 &
F30602-97-C-0067
PE - 63760E
PR -:F061 & F062
TA - 01
WU - 00

6. AUTHOR(S)

Stephen F. Smith and Karen L. Myers

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3890

SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025-3453

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington, VA 22203-1714

AFRL/IFTD
525 Brooks Road
Rome, NY 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2002-62

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: John Spina, IFTD, 315-330-4032, spinaj@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

This document constitutes a joint final report by Carnegie Mellon University (CMU) and SRI International for their collaborative work on the JFACC program. The main accomplishment on the project was the development of an integrated planning and scheduling capability (the JFACC Planner/Scheduler, or JPS) that supports generation of tightly linked air operations plans and schedules, as well as their adaptation in response to changing tasks and resource availability. This effort built on existing planning (from SRI) and scheduling (from CMU) technologies that provide core generation and repair techniques, along with corresponding knowledge bases for air operations for each technology.

14. SUBJECT TERMS

air operations, planning and scheduling, knowledge bases

15. NUMBER OF PAGES

68

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

.

Table of Contents

List of Figures

List of Tables

1. Introduction

This document constitutes a joint final report by Carnegie Mellon University (CMU) and SRI International for their collaborative work on the JFACC program under contracts F30602-97-2-0066 (CMU) and F30602-97-C-0067 (SRI).

The main accomplishment on the project was the development of an integrated planning and scheduling capability (the *JFACC Planner/Scheduler*, or *JPS*) that supports generation of tightly linked air operations plans and schedules, as well as their adaptation in response to changing tasks and resource availability. This effort built on existing planning (from SRI) and scheduling (from CMU) technologies that provide core generation and repair techniques, along with corresponding knowledge bases for air operations for each technology.

By *planning*, we refer generally to the process of deciding *what* to do; that is, the process of transforming strategic objectives into executable activity networks. We use the term *scheduling* to refer to the process of deciding *when* and *how*; that is, which resources to use to execute various activities and over what time frames. Traditionally, planning and scheduling have been viewed as distinct activities, and different solution techniques and technologies have emerged for each. Relatively few attempts have been made to combine these technologies into larger integrated frameworks.

Our technical work focused on defining techniques that provide increased coupling between the planner and scheduler. Our expectation was that tighter coupling would yield performance gains along three dimensions: (1) reduced overall generation and repair times, (2) improved quality of plans and schedules and (3) greater stability in the solutions generated over time. The particular approach that we pursued was based on the idea of approximating the resource requirements of different planning options using a model of *resource intensity*, and incrementally exchanging and exploiting information about likely resource shortfalls and excess capacity to identify options that best utilize available resources.

Our work progressed in four phases. First, we developed a baseline integration of our existing planning and scheduling technologies that was organized around a simple *waterfall* model of interaction. With this model, planning and scheduling proceed in sequential, lockstep fashion; any problem encountered during scheduling simply triggers the generation of a new plan. Second, we explored a simple *single-dimensional model* of resource intensity that used qualitative assessments as the basis for reasoning about expected resource usage. Third, we extended our integration to support plan and schedule repair in response to changes in tasking and resource availability. Fourth, we explored a *multidimensional model* of resource intensity grounded in quantitative estimates of expected resource usage. This multidimensional approach models resources at a finer level of granularity than does the single-dimensional approach, and so would be expected to have improved predictive value for estimating resource usage.

Experimental evaluation was an integral part of our work. Our experimentation was split into three main areas:

- *Experiment A: Resource Feasibility Checking - Single-dimensional Intensity Model* This experiment focused on evaluating the extent to which the use of a simple single-dimensional intensity model to support resource feasibility checking during plan generation could improve computational performance without sacrificing quality.

- *Experiment B: Plan and Schedule Repair* This set of experiments tested a hypothesis similar to that of Experiment A, but for the case of plan and schedule repair rather than generation. In addition, it considered issues related to stability of plans and schedules as repairs were performed.

- *Experiment C: Tighter Coupling through a Multidimensional Quantitative Intensity Model* This experiment set consisted of five related experiments that evaluated the effectiveness of the multidimensional intensity model relative to both waterfall integration and the single-dimensional intensity model. The experiments tested the methods on a range of problems, as well as evaluating sensitivity of the key parameters of the multidimensional intensity adaptation method to small- and medium-size perturbations.

Our experimental results show that the intensity-based approaches produce plans of comparable quality to the waterfall integration method, but for greatly reduced computation time. The performance improvements tend to increase with the level of resource constrainedness of the problem, thus making our techniques most valuable in situations where resource contention is high. Our final set of experiments further showed that the multidimensional intensity approach provides significant improvement over the simpler single-dimensional model.

The results of this project have been documented in three published technical papers. The first paper addresses general issues for the problem of integrating planning and scheduling technologies. The second paper provides a brief overview of our intensity adaptation method, along with a small set of experimental results that validate its usefulness. The third paper provides a more detailed description of our approach, and includes a broader set of experimental results.

- *"Issues in the Integration of Planning and Scheduling for Enterprise Control"*, K.L. Myers and S.F. Smith, *Proc. DARPA Symposium on Advances in Enterprise Control,* 1999.

- *"Integrating Planning and Scheduling through Intensity Adaptation*", K. L. Myers, S. F. Smith, D. W. Hildum, P. Jarvis, R. de Lacaze, Proceedings of the IJCAI-01 Workshop on *Planning with Resources*, 2001.

- "*Integrating Planning and Scheduling through Adaptation of Resource Intensity Estimates*", K. L. Myers, S. F. Smith, D. W. Hildum, P. Jarvis, R. de Lacaze, *European Conference on Planning*, 2001.

The remainder of this report is organized as follows. Section 2 motivates the need for improved techniques for integrating planning and scheduling. Section 3 provides background information on the planning and scheduling technologies from SRI and CMU, along with an overview of the JPS system. Section 4 discusses key aspects of the air operations domain and their impact on the design of algorithms for integrating planning and scheduling. Section 5 describes the single-dimensional intensity model and and its use for supporting resource feasibility checking in plan/schedule generation, along with the results for Experiment A. Section 6 describes our approach and experimental evaluation for plan/schedule repair (i.e., Experiment B). Section 7 summarizes our multidimensional intensity model along with the results for Experiment C. Section 8 presents our conclusions for the project.

Our initial experiment plan called for the connection of the planner/scheduler to a simulation environment, as a means of enabling evaluation of the embedded and real-time capabilities of the system. Two candidates were considered: the Air Operations Enterprise Model developed by BBN, and the SimFlex simulation framework developed by SRI in the previous phase of the JFACC program. In consultation with program management in mid-summer 2000, it was decided that we should concentrate our effort on increasing the sophistication of our techniques for planner/scheduler interactions, with an eye toward transition of the technology to operational programs. For the sake of completeness, the Appendix includes an outline for an experiment to evaluate the effectiveness of JPS as an embedded controller.

2. The Need for Integrated Planning and Scheduling

Goal-oriented activity in complex domains typically requires a combination of planning and scheduling. Military planners must select courses of actions that achieve strategic objectives, while making the most of available assets. A manufacturing facility must develop process plans for ordered parts that can be integrated cost-effectively with current production operations. Space observatories must allocate viewing instruments to maximize scientific return under a large and diverse set of causal restrictions and dependencies. Though conceptually decomposable, planning and scheduling processes in such domains can be and often are highly interdependent. Different planning options for achieving a given objective can make quite different demands on system resources; correspondingly, current resource commitments and availability will impact the feasibility or desirability of various planning options. The dynamics of the operating environment complicate matters further, requiring efficient response to continual unexpected changes to system objectives and resource availability.

The effectiveness of goal-oriented activity is ultimately tied to an ability to keep pace with evolving circumstances, and one recognized obstacle in practice is poor integration of "planning" and "scheduling" processes. In manufacturing organizations, this problem has been characterized as the "wall between engineering and manufacturing". Similar sorts of barriers can be found in other large-scale enterprises. The crux of the problem is lack of communication. Plans are developed with no visibility of resource availability and operational status and, likewise, schedules are developed and managed without knowledge of objectives and dependencies. Without such information exchange, planning and scheduling processes are each forced to proceed in an uninformed and inherently inefficient manner. In the simplest case, the result is an *iterative waterfall* model of integration, where planning and scheduling are performed in sequential lockstep fashion, and any problem encountered during scheduling simply triggers the generation of a new plan. Unfortunately, the iterative waterfall approach to integration is quite common in many application domains, leading to artificially inflated lead times and costs in manufacturing environments, more protracted and more resource wasteful military campaigns, missed opportunities at scientific observatories, and so on.

Despite the broad need for better integration of planning and scheduling processes, research in this area has been rather sparse [Dean et.al 88, Muscettola et.al 92, McVey et.al 97, Sadeh et.al 98, Chien et.al 99, Jonsson et.al 00, Smith et.al 00, Srivastava et.al 01]. In many cases, the domains of interest have been heavily resource driven (e.g., [Muscettola et.al 92, Sadeh 98, Chien 99, Jonsson 00]), with simple, locally contained planning subproblems. By and large, these problems have not required sophisticated planning capabilities and have been solved using extended scheduling techniques. Other work has focused mainly on structured benchmark problems (e.g., [Srivastava et.al 01]), where systematic techniques can be tractably applied. Such methods do not scale to problems such as the air operations domain considered in our work.

3. The JPS System: Components and Design

This section describes the core planning (Section 3.1) and scheduling (Section 3.2) technologies used within the project, as well as the architecture of the JFACC Planner/Scheduler system (Section 3.3).

3.1. Planning Technology

The Continuous Planning and Execution Framework (CPEF) provides the basis for the planning component for our work (Myers 99). CPEF embodies a philosophy of plans as dynamic, open-ended artifacts that evolve in response to a continuously changing environment. CPEF provides a range of operations required for continuous plan management, including *plan generation*, *plan execution*, *monitoring*, and *plan repair*. Plan generation within CPEF is based on the Continuous Hierarchical Incremental Planner (CHIP) system. CHIP is a hierarchical task network (HTN) planner derived from SIPE-2 (Wilkins 88). CHIP is, essentially, a reengineering of the SIPE-2 system that supports the incremental planning model required for our planner/scheduler integration.

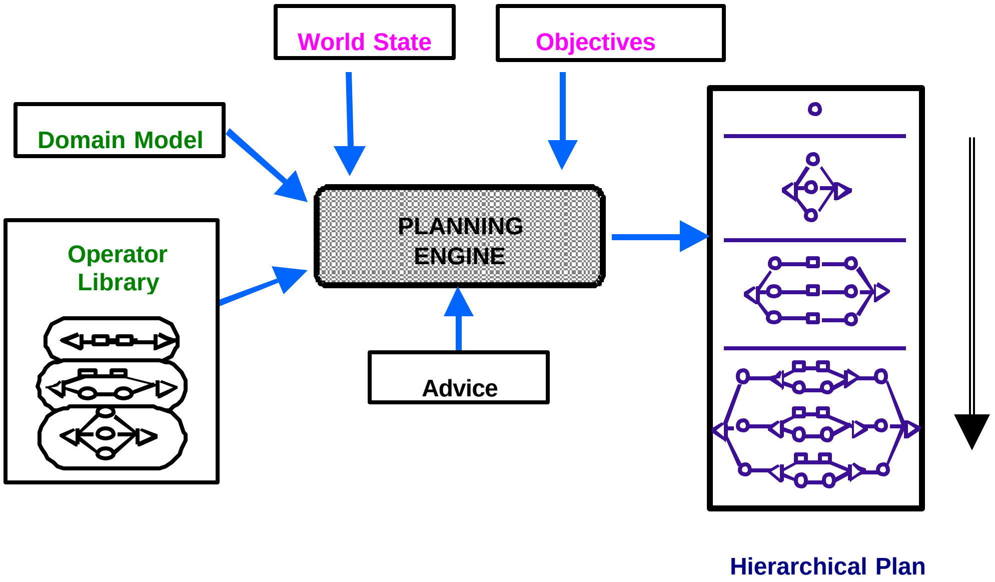

Figure 1. The CHIP Planner

Figure 1 illustrates the architecture of the CHIP system. The role of the planner is to expand an initial set of user-provided objectives to a complete set of actions for achieving those objectives. The core CHIP planning engine is domain independent; application of CHIP to a specific problem domain requires the formulation of three types of background information to inform the planning engine during its operation.

- The *operator library* provides the strategic knowledge needed to decompose objectives. Each operator describes one approach for decomposing a specific objective or task into a more detailed set of objectives, tasks, and actions. For example, the air operations domain contains a number of different strategies for neutralizing an enemy IADS. Constraints defined within an operator impose conditions on when individual operators can be used.

- The *domain model* describes general constraints about the application domain. For the air operations application, the domain model includes geographic information for the area of operations and general information about forces and equipment.
- The *world state* model describes the expected conditions in which the plan would be executed. For the air operations domain, the world state includes information about resource ava ilability, threats, and centers of gravity.

Given an initial set of objectives, the planning process proceeds in top-down fashion. At each level, the planning engine selects operators to refine outstanding objectives and tasks to the next level of detail. Much of the selection process involves validating the conditions of applicability for operators: those conditions must be satisfied relative to the world state and domain models in order for an operator to apply. The planning process terminates when all objectives and tasks have been refined to executable actions.

The CHIP system also folds in the capabilities of the Advisable Planner (Myers 96), which provides an advice-taking layer that enables a user to guide and direct the plan generation toward solutions that match his or her individual preferences. Advice enables users to express preferences for strategies with certain characteristics, or that use or avoid specified entities (such as certain resources) in designated situations.

3.2. Scheduling Technology

The base scheduling capability for our work is provided by ACS (Air Campaign Scheduler). ACS is an air operations scheduler constructed using OZONE (Smith, Lassila and Becker 96), a customizable constraint-based modeling and search framework for developing incremental scheduling applications. OZONE consolidates the results of application development experiences in a range of complex domains, including one recently deployed system for day-to-day management of airlift resources at the USAF Air Mobility Command (AMC) (Becker and Smith 00).

The ACS scheduler adapts techniques underlying the AMC application to the air operations domain. Its functional scope is depicted in Figure 2. ACS is provided with three broad types of inputs:

- a set of *demands*, relating to input tasks (targets/DMPIs) to be scheduled and their corresponding constraints,
- a set of *capabilities*, specifying (1) what types and amounts of resource capacity (assets, munitions) are available for use, (2) where they are positioned in theater and over what interval(s) they may be used, and (3) a table of weaponeering solutions, that map the effectiveness of different platform/munitions pairs to various target category codes,
- *world state* information, indicating such exogenous factors about the execution environment as threats and weather, as well as information relating to execution results.

The output from ACS provides inputs to feed generation of a Dynamic Air Execution Order (a “continuous” form of a Master Air Attack Plan). ACS generates a set of assigned strike missions designating, for each input target/DMPI demand:

- the set of sorties to be flown (possibly converging on the target from different bases),
- the numbers of aircraft and munitions to be expended from each base and,
- the precise time windows for various stages of the flight itinerary (including TOT windows).

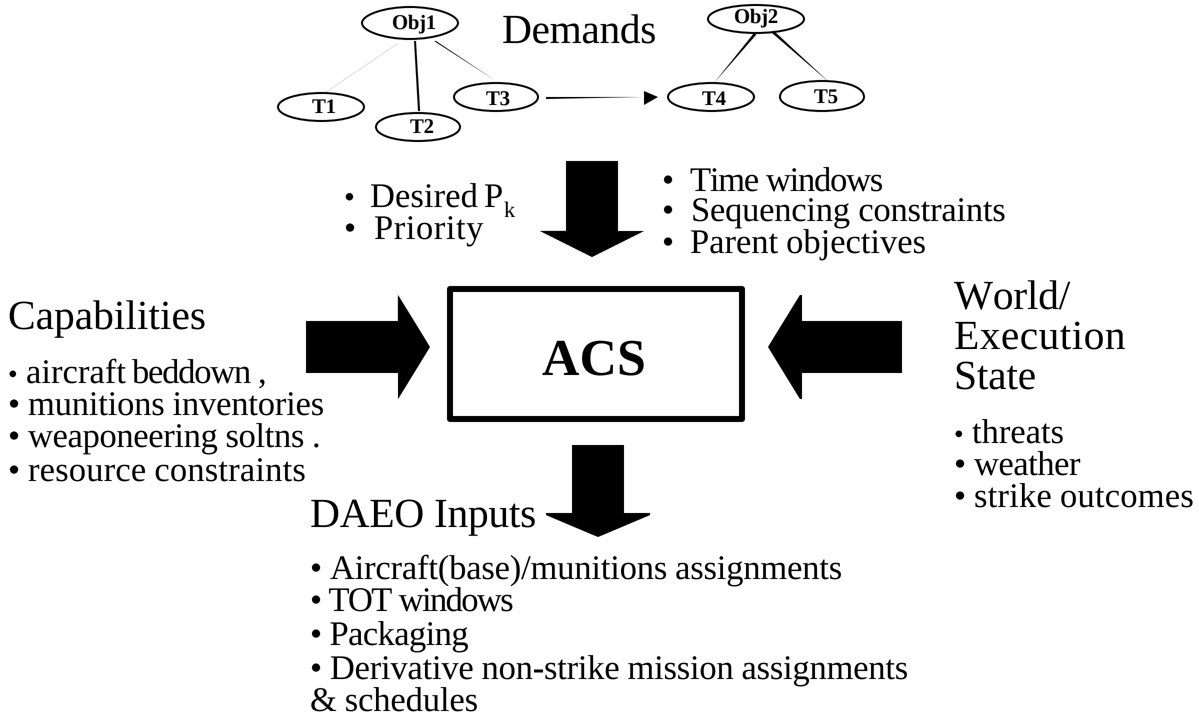

Figure 2: ACS Functional Scope

On its own, ACS provides a range of air campaign scheduling capabilities. In generative mode, it can be used to efficiently generate assignments of aircraft and munitions to a given set of input target/DMPI demands. As suggested above, these assignments take into account such considerations as target priorities, desired levels of destruction, time-on-target (TOT) windows, temporal sequencing constraints, feasible weaponeering solutions, and aircraft/munitions positioning and availability constraints. ACS can also be used in incremental mode, both (1) to accommodate and integrate new demands into a continuously evolving air campaign schedule, and (2) to reactively reallocate in response to unexpected changes in the execution status (e.g., loss of aircraft, insufficient destruction effect). Finally, ACS provides capabilities for selective (user-driven) relaxation of constraints, providing a basis for exploring alternatives (e.g., delaying missions, surging) in situations where all constraints cannot be satisfied. The ACS user interface (See Figure 3) provides a range of displays for visualizing and incrementally manipulating input constraints and allocation decisions.

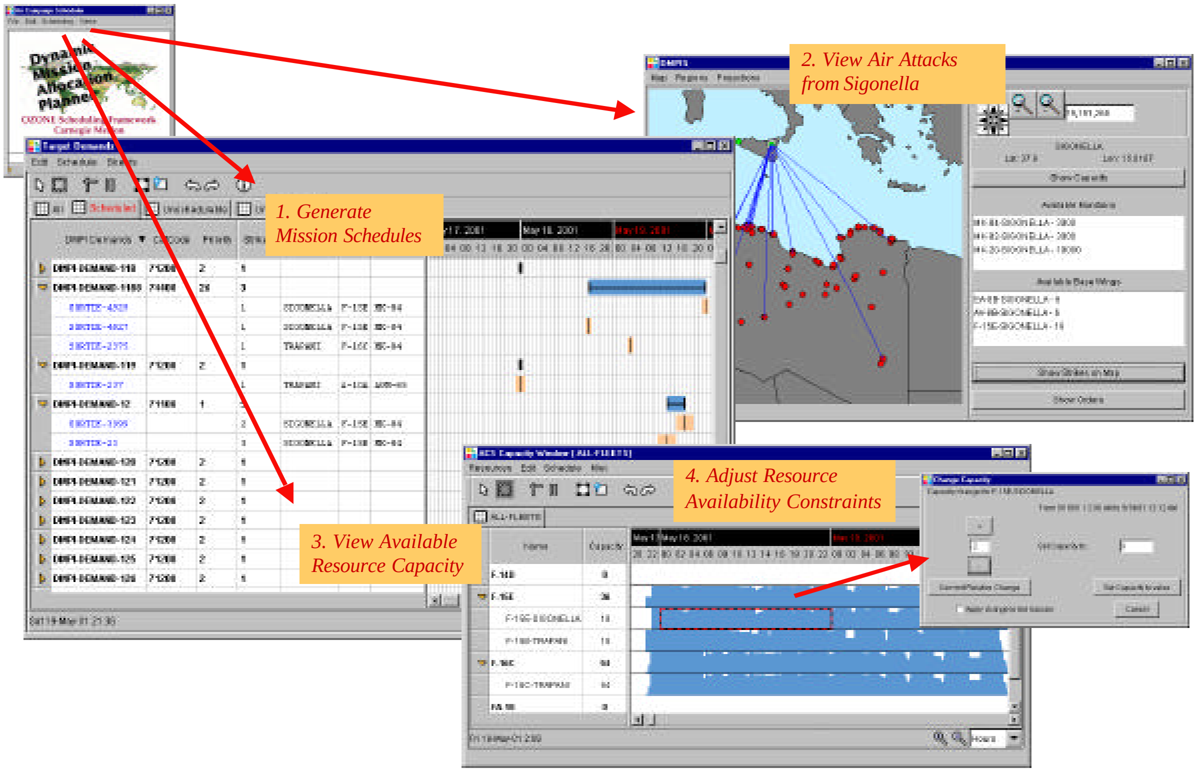
1. Generate Mission Schedules
2. View Air Attacks from Sigonella
3. View Available Resource Capacity
4. Adjust Resource Availability Constraints
Dynamic Mission Allocation Planner
OZONE Scheduling Framework
Carnegie Mellon
Target Demands
ACS Capacity Window [ALL-FLEETS]
Change Capacity

Figure 3: ACS Graphical User Interface

3.3. JPS Architecture

The JPS system, as illustrated in Figure 4, comprises three main components: the planner, the scheduler, and the plan server. These components exchange information through a set of JPS-specific protocols designed to support the exchange of plans, schedules, and status-related information. The actual communication of messages is provided by a KQML message-passing system, which provides peer-to-peer message exchange across the Internet (Finin et al. 92). We chose this design to enable SRI and CMU to run their technologies on local machines, thus providing a simple and natural environment for collaborative development.

The JPS plan server provides a central repository for plans and schedules, as well as for information related to the planning/scheduling process. The plan server accepts incoming information from agents, performs necessary processing, and stores relevant information in its internal representation. Information stored within the plan server (including scheduled plans) can be accessed through a rich query language.

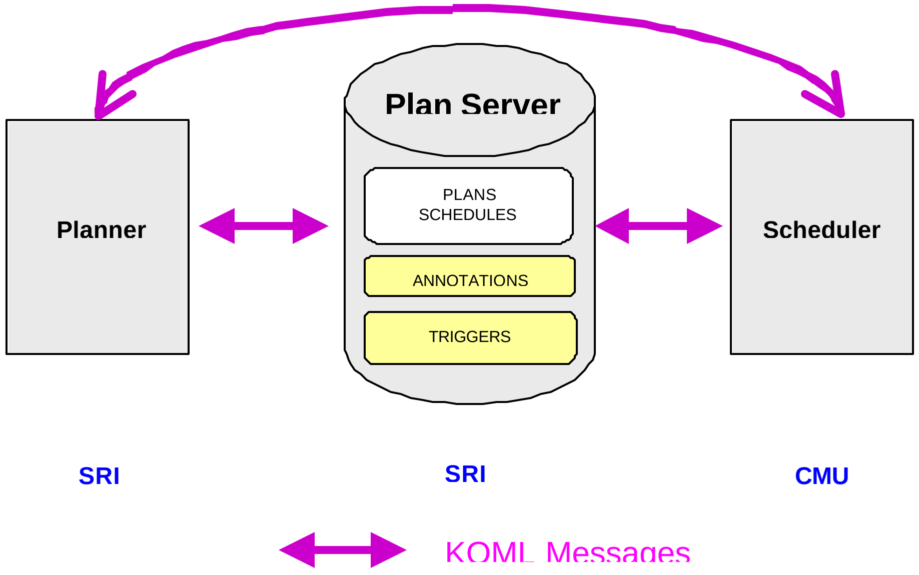

Figure 4. JPS Architecture

JPS has no centralized controller to dictate when planning or scheduling takes place. Rather, JPS employs a decentralized framework for managing the interactions between the planner and scheduler. The approach builds on two key mechanisms within the plan server: *annotations* and *triggers*. Annotations are declarations about the status of products (i.e., plans, schedules) and processes (i.e., the status of planning or scheduling operations), and are asserted into the plan server knowledge base. Triggers are event-response rules activated by the assertion of annotations, and result in messages being sent to designated recipients. Through appropriate declarations of annotations and triggers, components can initiate activities by other components without the need for a centralized controller.

For example, once the planner completes a given subplan, it posts an annotation of the form

(SUBPLAN-COMPLETE <subplan-id> <plan-id>)

to the plan server. The plan server includes a trigger that is activated by receipt of this annotation, which results in a message being dispatched to the scheduler requesting that it schedule the new subplan.

This distributed control model provides great flexibility, enabling interactions between the planner and scheduler to be added through the declarative specification of annotations and triggers rather than through code changes. This design was helpful with the evolution of JPS because it enabled quick and easy updates to control flow within the system. The design also promotes modularity: incorporation of a new component only requires 'wrapping' the technology to support the JPS communication protocols, and designing appropriate annotations and triggers. For example, a module that does plan evaluation could easily be integrated into JPS: the module would need only post a trigger saying that it was interested in being notified when subplans were completed so that it could retrieve the plans and assess them (possibly while scheduling is occurring).

The integration infrastructure within JPS builds substantially on components of the architecture of CPEF, which was developed by SRI during the previous phase of the JFACC program. The KQML message-passing capabilities were taken directly from the earlier system. The JPS plan server was built on top of the CPEF plan server, extending it to support the schedules, subplans, and subschedules. In addition, many of the communication protocols, annotations, and triggers from CPEF were incorporated into the JPS system; however, many extensions to these concepts were required to support the JPS model of incremental planning, as well as scheduler-related requirements.

4. The Air Operations Domain

Applications that require integrated planning and scheduling will have individual characteristics that dictate the relative importance of each of these capabilities. Much of the work to date on combining planning and scheduling has focused on *resource-driven* domains (such as satellite observation scheduling (Muscettola et al. 92)), which emphasize optimization of resource usage in satisfying a pool of tasks. In contrast, the air operations domain has a more *goal-driven* flavor: while effective resource usage is important, the key motivation is to identify and schedule actions that will ensure attainment of stated objectives.

Objectives within our model of the air operations domain reduce to goals of neutralizing enemy capabilities (e.g., antiaircraft capability, electricity production, communications) modeled as hierarchical networks that ground out at the level of specific targets. We provide several strategies for attacking different network types that vary in their aggressiveness, and hence resource demands. For example, strategies range from attacking all components in a network, to attacking a coherent subset, or an isolated node (Lee 98).

Resources (i.e., aircraft, munitions) are assigned to support prosecution of individual targets. For a given type of target, there are usually several alternative aircraft/munitions configurations that could be used. However, different configurations will have different degrees of effectiveness, and hence the numbers of resources that must be allocated to achieve the desired effect can vary with each choice. Quantities (or capacities) of different types of resources are positioned at various locations nearby or within the geographic region of interest. The set of resources assigned to fly against a given target can vary in type and, depending on availability, may either originate from multiple locations (converging on the target within a particular time interval) or recycle from the same base location (making sufficient sets of consecutive strikes on the target).

The style of planning required for this domain differs markedly from standard approaches to automated planning. Here, the search space is dense with solutions, making it easy to find a plan that satisfies stated goals. The real challenge is to find 'good' plans rather than settling for the first available solution. While most automated planning systems seek to minimize plan size, bigger plans tend to be better in this domain. For example, eliminating more of an enemy's missile sites tends to improve the quality of a plan for neutralizing enemy attack capability. Note that maximizing plan size is not equivalent to maximizing resource usage: the planner and scheduler must still decide how to allocate available resources economically to support potential activities.

Air operations commanders generally apportion a set of resources for a given set of high-level objectives; human planners are expected to develop solutions that maximize the likelihood of objective attainment while staying within the resource allotment. Our planning models incorporate this *apportionment perspective* into their design. In particular, initial plans seek to capitalize on all available resources; as resource problems arise, strategies are adopted that decrease resource usage.

5. Experiment A: Resource Feasibility Checking – Single-Dimensional Intensity Model

This section describes an initial integration experiment to evaluate the extent to which incremental resource feasibility checking improves the efficiency of plan/schedule generation. More specifically, we first investigate the extent to which incorporation of a simple form of resource feasibility checking during plan generation can improve the computational performance of generating a combined plan/schedule without sacrificing plan/schedule quality. The description provided below conforms to the documentation format provided by the JFACC program.

5.1. Statement of Experimental Objectives

5.1.1. Hypotheses

Hypothesis: Early consideration of resource allocation issues during planning can improve efficiency substantially.

Level: Process Experiment

Description: To test this hypothesis, we will run the integrated planner/scheduler in two modes.

- For the baseline, we will provide a loosely coupled integration of planning and scheduling, in which the planner generates a plan in isolation and then passes it to the scheduler. The output from the planner will consist of a combination of Combat Air Patrol (CAP) missions and targets, along with sequencing, priority and mission horizon information. The scheduler will perform resource allocation and time-on-target assignments for the provided plan. Should the scheduler be unable to produce an acceptable schedule, the process of plan generation followed by scheduling will be repeated. We refer to this baseline as the *iterative waterfall* mode.

- The *resource feasibility checking mode* will be similar to the baseline but it will support intermediate, incremental invocation of the scheduler by the planner, during planning, to assess the feasibility of potential strategies under consideration.

Various approaches can be considered for feasibility checking during planning to determine the viability of current decisions. The approach taken for this experiment will be based on a simple single-dimensional model of resource intensity.

The experiment will compare the time required to generate a final scheduled plan by each of these planning/scheduling modes.

5.1.2. Value

Effective command and control requires both timely response to unexpected events and consideration of a broad range of options. By making substantial reductions in the time required to develop and adapt quality plans and schedules, the proposed resource feasibility-checking techniques will enable commanders to rapidly develop and evaluate multiple courses of action in highly dynamic settings.

5.2. Description of Experimental Setup

5.2.1. Simulation Features

This experiment did not require a simulation environment, and was conducted independently of program-wide Enterprise Models.

5.2.2. Variables or Correlated Parameters

Independent variables: resource availability profiles.

5.2.3. Specification of a Set of Test Runs

The improvements yielded by feasibility checking can be expected to vary depending on the degree of resource constrainedness of the underlying problem: greater benefits should result for problems where more strategies are resource infeasible within a given mission time horizon. The experiment will take this factor into account by considering resource availability profiles that provide qualitatively distinct points along the spectrum of resource constrainedness. For several points along this spectrum, we will define a distribution of resource profiles. Multiple runs will be performed for each point by selecting randomly among the resource profiles in the corresponding distribution.

In any given run of the system in baseline mode, replanning will be triggered in the event that the scheduler is unable to generate a resource-feasible schedule that achieves all requested objectives within the specified mission time horizon. In resource feasibility checking mode, indication of infeasibility will divert the planner to consider other strategic alternatives. System performance in both modes will be measured in terms of the computation times required for planning/scheduling across various test runs.

5.3. Results

5.3.1. Overview of the Single-dimensional Intensity Approach

As noted above, one can consider a variety of strategies for providing resource feasibility checking during planning, based on when feasibility is assessed, the nature of the feedback from the scheduler, and the corresponding response. This section reports results for an approach that assesses feasibility at the level of *subplans* for the overall set of objectives, using a model of *intensity* to approximate resource demand, and *adaptation* by the planner in response to scheduler feedback.

The approach involves planning in strategy-to-task fashion down to a specified level of detail (the *decomposition layer*), and then splitting into subplans that are planned separately (see Figure 5). The decomposition layer, defined implicitly in terms of specific goals, separates the higher-level strategic decisions that define overall plan structure from the planning of (mostly independent) lower-level objectives.

After completion of each subplan, the scheduler performs incremental resource allocation for the actions introduced by the subplan, relative to resource assignments made for previous subplans. In the event that the scheduler is unable to produce a satisfactory resource assignment, the planner will modify a completed subplan to reduce resource demand, and then forward the revisions to the scheduler for appropriate adjustments to the current schedule. Once all outstanding resource problems have been resolved, the planner continues with generation of remaining subplans until completion of a full plan and schedule. With this approach, the integrated plan and schedule is built in piecewise, incremental fashion, with adjustments made in response to detected resource problems.

This incremental approach would be ineffective for domains in which extensive strategic dependencies link objectives. However, in our models for the air operations domain, most dependencies occur at the level of resource allocation, thus enabling the separation of the planning for individual objectives. The incremental approach has the added benefit that it can be used for dynamically extending plans to include additional objectives as plan execution unfolds.

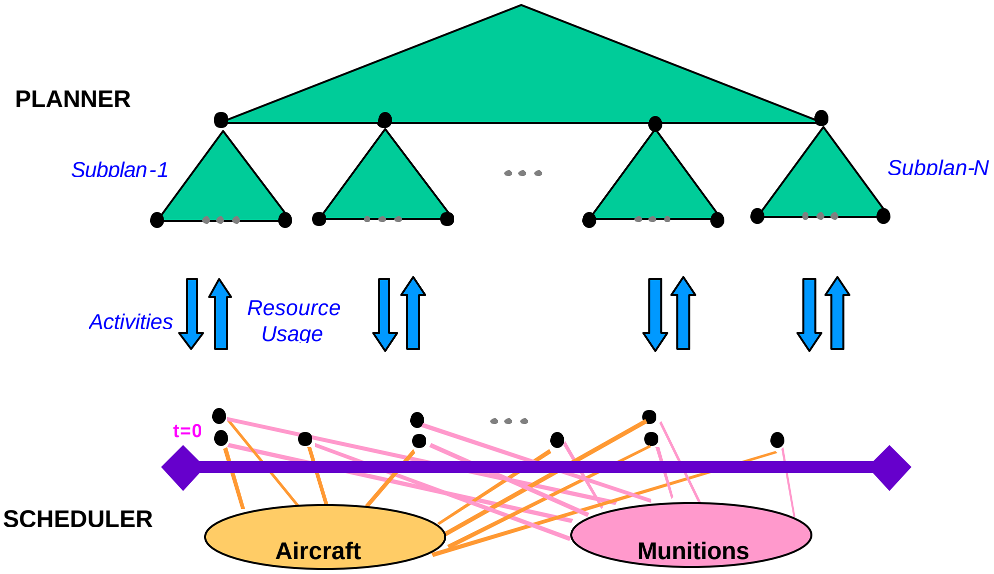

Figure 5. Planner/Scheduler Control Flow

To make informed decisions about its choices, a planner requires some model of the resource impact of its decisions. Although the specific actions that require resources do not appear until the lowest levels of our hierarchical models, high-level decisions have a great impact on resource requirements. For example, the decision of whether to employ a passive or more proactive approach to defending assets will greatly influence resource requirements, although the actual missions that require resources are planned at much lower levels of detail. For this reason, our approach to linking planning and scheduling builds on a heuristic characterization of expected resource usage by a planning operator, which we refer to as an operator's *intensity.*

For this first experiment, we employ a simple model of operator intensity that represents a qualitative assessment of the operator's expected resource usage relative to alternatives for the same task. The air operations domain, for example, contains multiple operators for neutralizing an enemy's communication capability, ranging from taking out a single site, to destroying some select subset of communication devices, to eliminating all communication nodes. For an intensity scale of [0 40] (as was used in our experiments), the first operator might be ranked a 10, the second a 20, and the third a 40 to reflect their relative levels of expected resource consumption.

Such qualitative ratings should be readily assessable by the knowledge engineer who develops the planning operators, especially since they need only be approximate.

The incorporation of intensity information to guide planning occurs at the level of subplans. For a given subplan, the planner calculates a *target intensity*, denoted by I^T. This value represents the expected 'ideal' use of resources for a particular subplan, relative to availability and expected demand for remaining subplans. When faced with a choice among multiple applicable operators for a task in a given subplan, the planner will select the one that is closest to the target intensity for the subplan without exceeding it. During the plan/schedule development process, scheduler feedback could indicate a shortage/excess of remaining resources, relative to the subplans yet to be generated and scheduled. Such a shortage/excess would be reflected in the setting of the next target intensity at a lower/higher level; the planner would then be biased toward selecting operators with lower/higher intensity values to reduce/increase resource consumption levels. In this way, the planner dynamically adjusts its decision-making in response to scheduler feedback.

i - # of completed subplans

R^i - Remaining capacity after planning i subplans

n - # of subplans that require resources

K - Maximum intensity

Remainder per subplan: $\frac{1}{n-i} \times R^i$

Allocated per subplan : $\frac{1}{n} \times Capacity$

Usage ratio of remainder to allocated: $U = \dfrac{\frac{1}{n-i} \times R^i}{\frac{1}{n} \times Capacity}$

Target Intensity

K if $U >= 1$ (resource underutilization)

$U*(K\text{-}Delta)$ if $U < 1$ (resource overutilization)

Figure 6. Target Intensity Selection

Figure 6 defines the method used to compute target intensities for subplans, which provides the basis for the intensity adaptation approach. The value U corresponds to the ratio of available resources for the current subplan to the value originally allocated. In the case where $U>=1$, resource usage is 'on track' and the target intensity is set to the maximum value. In the case where *U<1*, resources have been overused. In this case, the target intensity is set to the product of *K - Delta* and the scaling factor *U*. Here, the value *Delta* can be set to cause more or less conservative back-off strategies in the face of resource over-utilization. For our experiments, intensity values lay in the interval [0,40], with Delta set to 10.

5.3.2. Experiment A Results

The metric for this experiment was generation time. One difficulty with this metric is that generation time can be minimized by always generating the simplest possible plan (since it would require the fewest resources). To counter that possibility, we adopted a plan generation strategy that assumes that the use of available resources should be maximized. Thus, initial plans seek to capitalize on all available resources; as resource problems arise, strategies are adopted that back-off on expected resource usage. This model corresponds to a situation where a commander has been apportioned a collection of resources, and is allowed to use them to the extent possible to accomplish his objectives. This *apportionment perspective* makes it possible for generation time to be a meaningful metric. Models of plan quality could have been used as an alternative mechanism; however, the development of adequate models of plan quality is an extremely challenging task.

The experiment involved comparing the computation time of the iterative waterfall and incremental intensity adaptation methods over a range of resource profiles. To draw fair comparisons with the intensity-based approaches, the waterfall method considers operators in decreasing order of intensity. This strategy generally yields a plan that is close to the largest supportable for the available resources but is not necessarily optimal (i.e., the chronological backtracking used by the waterfall method will stop when it finds the first solution, even though undoing an earlier operator choice might enable subsequent choices that are more aggressive).

Resource Profile		Munitions	Waterfall		Incremental	
			Time	Action	Time	Action
Max-80	100%	100%	175	295	232	295
Med-100	86%	105%	160	295	209	270
▨	▨	▨	▨	▨	▨	▨
Max-67	67%	60%	*	*	634	123
Min-100	12%	4%	*	*	613	62

Figure 7. Experiment A: Comparison of Plan/Schedule Generation Times (in seconds) for Iterative Waterfall vs. Incremental Methods

The most generous resource profiles used in the experiment contained sufficient resources for a plan in which maximum intensity strategies were always selected (the *maximum intensity plan*); successively smaller profiles contained fewer resources, with the smallest providing just enough resources for the plan for which the minimum intensity strategies were always selected (the *minimum intensity plan*). The maximum intensity plan contains 346 actions to schedule, while the minimum contains 62. These actions are mostly force application to designated DMPIs, but also contain small numbers of CAPs and jamming activities.

Highlights of the experimental results are displayed in Figure 7. The results showed that in cases where sufficient resources were available for the maximal plan, the waterfall method outperformed the incremental method with respect to generation time (on average, 180 seconds versus 235 seconds). In particular, the incremental method required approximately 30% overhead. However, in the more resource-constrained cases, the waterfall method failed to find any solutions after 10 hours of runtime, while the incremental method completed in times varying from 613 to 634 seconds (depending on the resource profile).

The behavior of the waterfall method in resource-constrained cases was far worse than anticipated. Analysis of test runs revealed that the large times resulted from the nature of the distribution of choice points. In particular, while numerous choices are available at low levels of plan decomposition, modifications at that level have minimal impact on resource requirements. Changes higher up in the planning structure are required to significantly impact resource requirements. Within the planner, backtracking proceeds in chronological order, resulting in lowest-level decisions being reconsidered before high-level ones. Thus much backtracking is wasted on making modifications to plans that have minimal impact on schedulability. In contrast, because the intensity-based method recomputes entire subplans, it is able to reduce resource requirements much more quickly.

One surprising aspect of the experiment runs was the much larger than anticipated runtimes for the incremental planning process. Additional analyses showed that much of the performance penalty stems from some inefficiencies in the implementation of the incremental planning method. Our HTN planner was designed originally to operate in a level-oriented mode: given a partially refined plan, it searches through the plan for all unresolved goals and then applies an operator to expand each to a set of more refined tasks. The basis for our integration with the scheduler, however, revolves around the notion of *subplans,* which constitute localized components of the overall plan. In particular, the system plans (and then schedules) individual subplans in turn. To facilitate rapid development early in the project, we decided to implement our subplan model of planning by having the planner simply 'copy down' those components of a given planning level that are not within the current subplan under consideration. Doing so enabled us to reuse much existing code. However, as we have discovered, the copying leads to the creation of much excess structure in the plan that significantly slows operation.

For example, for the test case used in Experiment A, level-by-level expansion requires a total of 20 planning levels and 28,858 planning nodes, while the incremental approach requires 44 levels and 38,424 nodes. This substantial increase in the overall size of the plan structure has caused the planning times to be much higher than is warranted for the approach.

This problem is an implementation issue only, not an inherent limitation of the incremental approach. Much of this extra cost could be eliminated through a reimplementation of the core planning algorithm in which subplans are generated ‘in place’, rather than repeatedly recopying the structure outside of the subplan.

In summary, our initial experiment showed that while the incremental mode incurs a performance penalty when there is an abundance of resources, it provides far superior performance when resources are limited. Furthermore, a better implementation of our incremental planning method would result in greatly improved performance.

5.3.3. Future Directions

This experiment can be extended in several directions. First, sensitivity analyses should be performed to provide a better understanding of how much the approach depends on the intensity assignments. Second, we believe that even better results could be obtained through planning strategies that embody a deeper understanding of resource allocation issues. Experiment C (Section 7) addresses both of these issues.

A third area for consideration relates to a more general control model. In particular, it would be valuable to consider a model that enables temporal deadlines (i.e., the mission time horizon) to be traded against intensity.

6. Experiment B: Plan and Schedule Repair

This section describes a second integration experiment designed and performed to evaluate the potential advantage of interleaved planning and scheduling for plan and schedule repair rather than generation. In addition, this experiment considers issues related to stability of plans and schedules as necessary solution repairs are performed. The description provided below conforms to the documentation format provided by the JFACC program.

6.1. Statement of Experimental Objectives

6.1.1. Hypotheses

Hypothesis B1: Incremental plan and schedule repair methods will improve the efficiency with which plans and schedules are adapted to account for new external events and the continuity (or stability) of plans and schedules generated over time (at the possible expense of some loss in overall solution quality).

Hypothesis B2: Tight coupling of plan and schedule repair will enable faster adaptations and better quality results in a given decision time frame than does loose coupling, where degree of coupling is characterized in terms of amount of information communicated.

Level: Process Experiment

Description:

When unexpected events occur or assumptions are invalidated, modifications to plans and schedules will be required. We are developing techniques that enable the planner and scheduler to guide the modifications made by the other, as a way to both reduce the time required to complete the adaptations, and to maintain greater stability (i.e., fewer changes) to currently planned and scheduled activities.

As a first step toward evaluating these techniques, we consider the prerequisite issue of incremental versus regenerative response to a new unexpected event. At both the planning and scheduling levels it is possible to consider modification approaches that either (1) recompute the plan or schedule from scratch, taking into account the new information or (2) incrementally revise (or repair) portions of the plan or schedule to account for the new information. Since the techniques we are developing for tighter coupling between planner and scheduler actions assume an incremental (or repair-based) approach to modification, it is useful to first quantify the tradeoffs surrounding this basic assumption. Accordingly, we will first compare regenerative and incremental modification strategies at both planning and scheduling levels for responding to specific events.

Given the results of this first step, we will compare the performance of repair techniques designed to communicate and exploit additional information (tight coupling) to a repair approach in which there is minimal communication between the planner and scheduler. In particular, the planner will communicate only plan changes and the scheduler only those actions that are no longer supported in the modified schedule in the "minimal communication" configuration.

The experiment will encompass repair tasks triggered by two types of events:

- changes in *tasking*, in particular the introduction of new objectives

– changes in *resource availability*, in particular the loss of some amount of resource capacity over some time interval

6.1.2. Value

Responsiveness to situation dynamics is critical for successful air operations. Adaptations to plans and schedules must be rapid and should minimize disruption to currently planned or ongoing activities. The development of techniques that can support faster, less-disruptive adaptations will enable more flexible and robust command and control.

6.2. Description of Experimental Setup

6.2.1. Simulation Features

This experiment does not require a simulation environment, and as such will be conducted independently of program-defined Enterprise Models.

6.2.2. Variables or Correlated Parameters

Independent variables: type and magnitude of changes, level of resource contention.

6.2.3. Specification of Test Runs

Different types of change can lead to widely varying impacts on a plan and schedule. For this reason, we will perform test runs that begin with an existing plan/schedule in place and involve responding to a set of different types of externally imposed changes (e.g., new objectives or guidance, updated resource availability profiles.). For each type of change, we will define multiple runs by varying the magnitude of the particular change. This will allow evaluation of tightly and loosely coupled revision strategies under varying assumptions as to the degree of disruptiveness to the existing plan and schedule.

We will measure comparative performance with respect to (1) the computational cost of the replanning and rescheduling processes, (2) appropriate measures of solution quality (e.g., overall makespan of operation, percentage of unsupportable missions), and (3) extent of change to current solution (or equivalently the degree of continuity in the plan/schedule).

6.3. Pre-Lab Analysis

It is likely that different plan and schedule repair strategies will be relevant to different types of externally imposed changes. Preliminary experimentation will be performed to tune and assess the prospective strengths of different schedule and plan repair strategies, and to determine system test configurations for loosely coupled and tightly coupled plan and schedule repair modes.

6.4. Results

6.4.1. Experiment B1: Response to New Objectives

This section reports the results of the study designed to quantify the tradeoffs between incremental and regenerative approaches to modifying the plan and schedule.

To this end, a set of problems requiring planner/scheduler response to the addition of new objectives to the original plan is defined. Using a basic model of the Cyberland Scenario and starting with an initial preexisting plan/schedule accounting for 253 DMPIs in place, four additional objectives of increasing size were defined as a set of alternative triggering events, respectively adding 0, 8, 25, and 86 new DMPIs to the overall plan. For each new objective, resource capacity profiles were systematically varied to provide a continuum of problem instances ranging from low contention (100% level – specified as a level of assets sufficient to accommodate all DMPIs in the original plan plus the largest new objective) to high resource contention (50% level). The set of resource profiles used for various aircraft and munitions levels is shown in Figure 8. For each of these problems, the set of tasks (DMPIs) reference a total of 11 different "catcodes" (or target types), and a given catcode has 4 to 8 distinct weaponeering solutions (feasible aircraft/munitions pairs). For each problem instance, a 2-day scheduling horizon was assumed. Any tasks that cannot be accomplished in this time frame are dropped as unsupportable. Hence, solution quality for this set of experiments is considered to be a function of the number of tasks that are ultimately supportable.

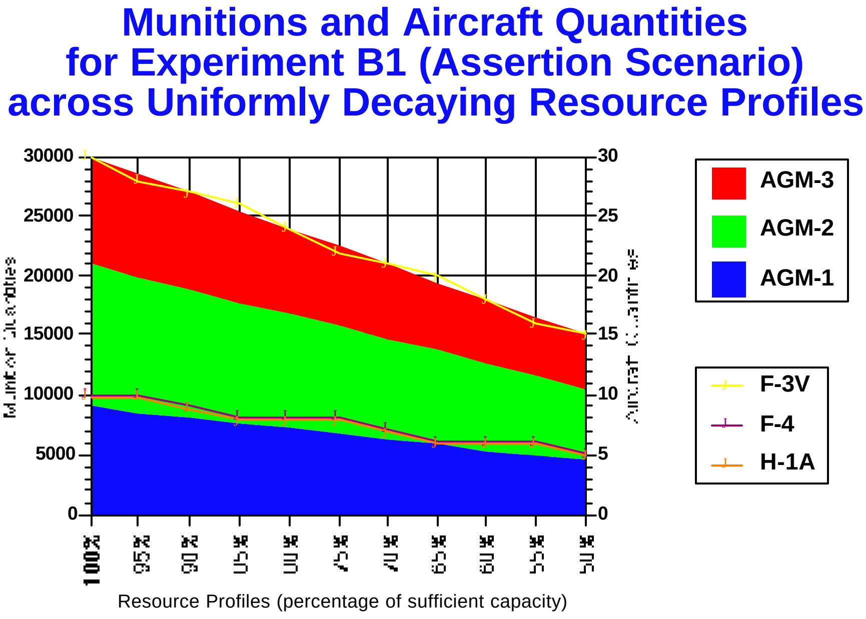

Figure 8. Resource Profiles for Experiment B

In responding to a request to augment the plan to incorporate a new objective, the planner (in incremental plan repair mode) invokes a two-stage process:

> First, an analysis is performed to determine which (if any) new objectives need to be added to the plan. The new objective may be already accounted for in the current plan – this is in fact what happens in the case of the smallest objective defined above (yielding 0 new tasks).
>
> Second, subplans are expanded for any added objectives.

Since the planner will produce the same final plan in both regenerative and incremental mode, the only interesting point of comparison at this level is relative computation time. Figure 9 gives these results for each of the four new objectives. As can be seen, the incremental approach is superior across all four problems despite the fact that there is some overhead to performing the analysis step. The incremental repair approach is a clear win, and there seems little reason to consider a regenerative strategy to modification.

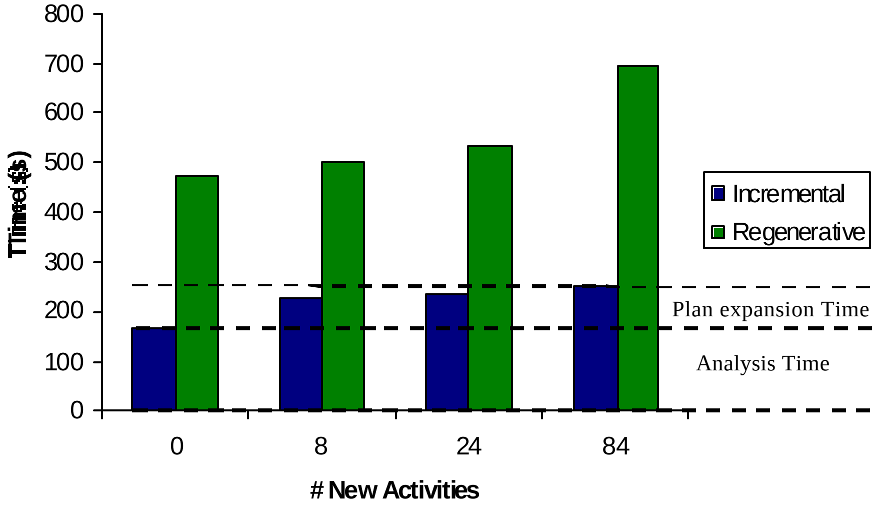

Figure 9. Computation Results at Planning Level

The tradeoff between incremental and regenerative is not as clear cut at the scheduling level. Since the scheduler is optimizing heuristically, there will most likely be additional distinctions in performance from solution quality and solution stability standpoints. For purposes of this study, two different heuristic strategies are considered as a basis for determining the "most critical" task to schedule next:

- Fewest resource alternatives first – this strategy prefers to next schedule the task that has the fewest number of resource alternatives (in this case corresponding to the task with the fewest number of weaponeering solutions).
- Largest task first – this strategy prefers to next schedule the task that has the largest resource requirement (in this case corresponding to the task that will require the largest number of strikes).

Each of these strategies is initially run in two different modes (a third mode will be introduced later):

- Regenerative – the current schedule is deleted and the entire scheduling problem is re-solved after incorporating any new tasks produced by the planner in response to the new objective.
- Incremental – the current schedule is kept intact, and new tasks introduced by the planner are scheduled (if possible) using existing excess resource capacity.

Figure 10 considers the computational tradeoff across different problems and resource contention levels. In these and subsequent graphs, the headings Assertion 2, Assertion 3 and Assertion 4 correspond respectively to new objectives 2, 3, and 4 given to the planner. (Since objective 1 yields no new tasks there is no rescheduling problem.) The following observations can be made:

— With sufficient excess capacity, the incremental strategy is always more efficient.

— As the size of the plan addition increases, the efficiency advantage of the incremental strategy diminishes.

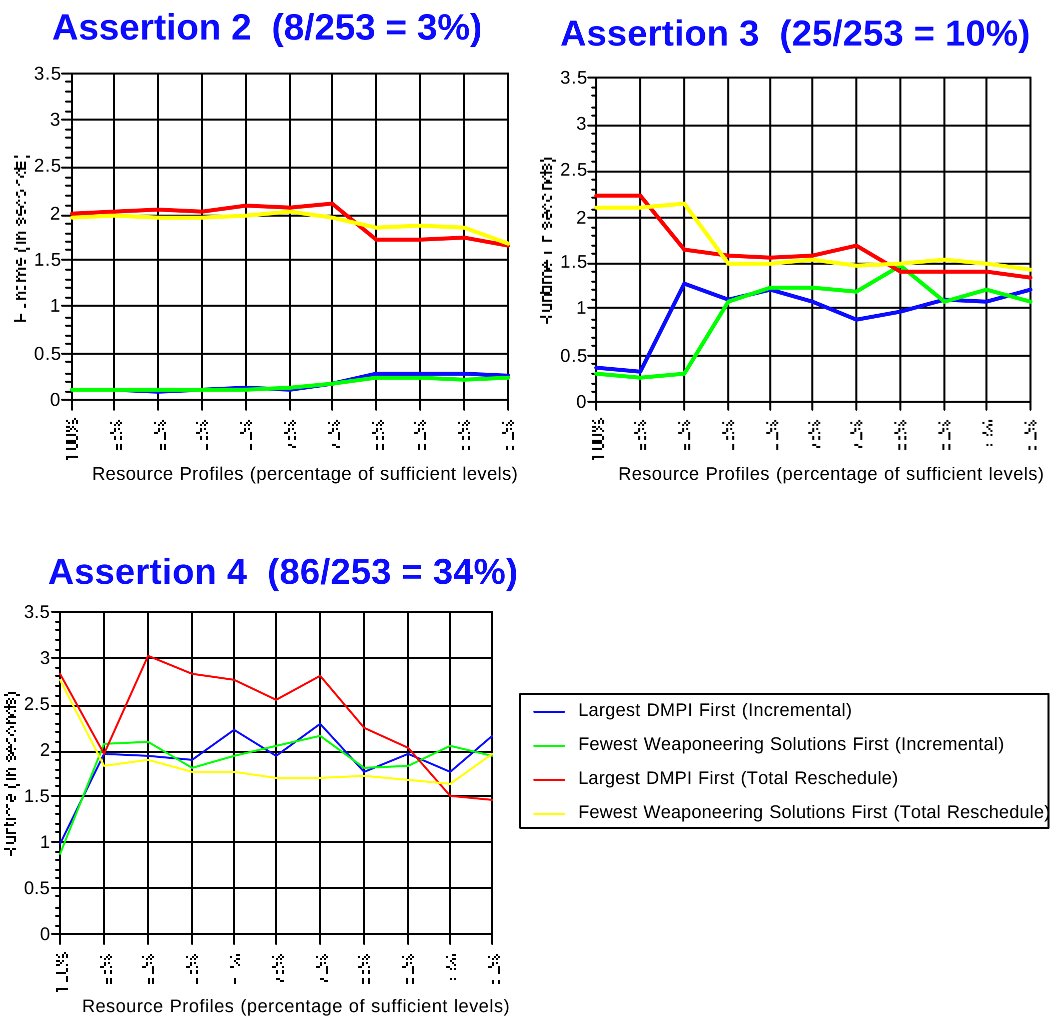

Figure 10. Computational Tradeoff between Regenerative and Incremental Scheduler

Figure 11 shows comparative results from the standpoint of solution quality (in this case percentage of tasks that are supportable). No significant difference can be seen between incremental and regenerative strategies along this dimension (and, in fact, in a few cases the incremental strategy outperforms), but there is perhaps insufficient variability in problem structure across this set of problems (since all problems derive from the same base plan) to draw any strong conclusions here. With respect to heuristic strategies, the results indicate that fewest resources (i.e., weaponeering solutions) first outperforms as resource contention gets higher.

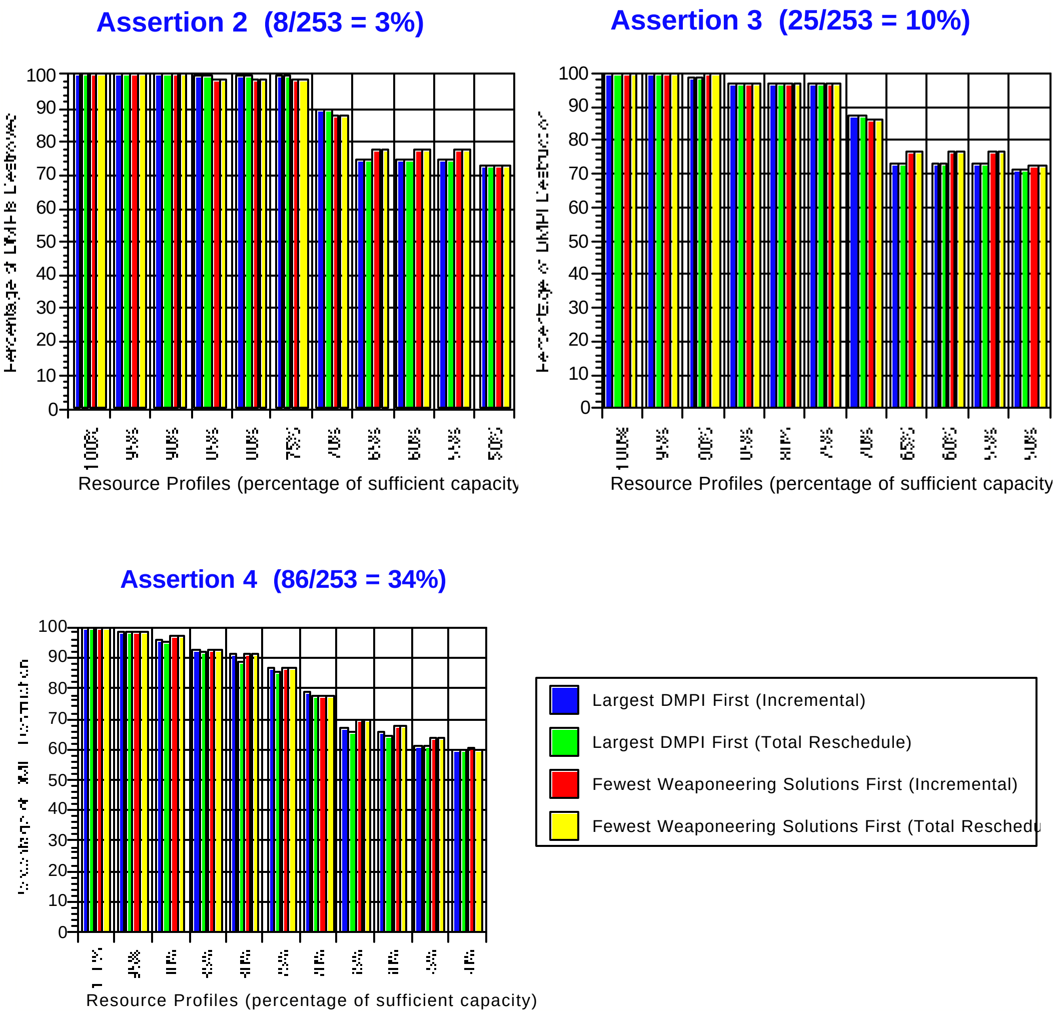

Figure 11. Solution Quality Tradeoff between Regenerative and Incremental Scheduler

Finally, Figure 12 and Figure 13 give solution stability results for both heuristic strategies. The results show the number of changes to the initial schedule (scheduled times, reassigned resources, and alternative weaponeering solutions) that result from responding in regenerative mode. Recall that the incremental strategy keeps the solution intact and makes no changes. One interesting fact to note is that the fewest weaponeering solutions first strategy appears to be much less disruptive than largest task (DMPI) first.

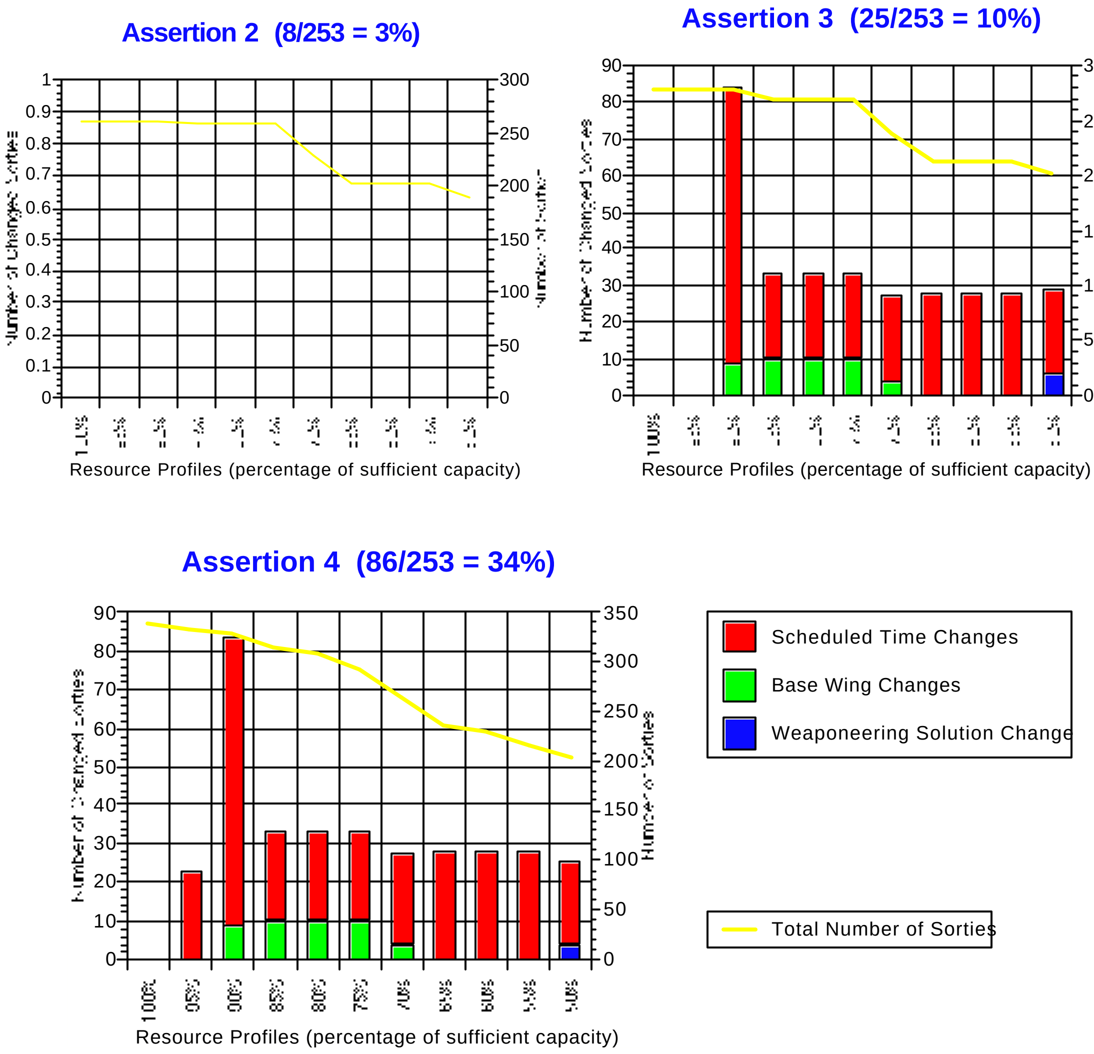

Figure 12. Changes to Initial Schedule – Regenerative, Fewest Weaponeering Solutions First

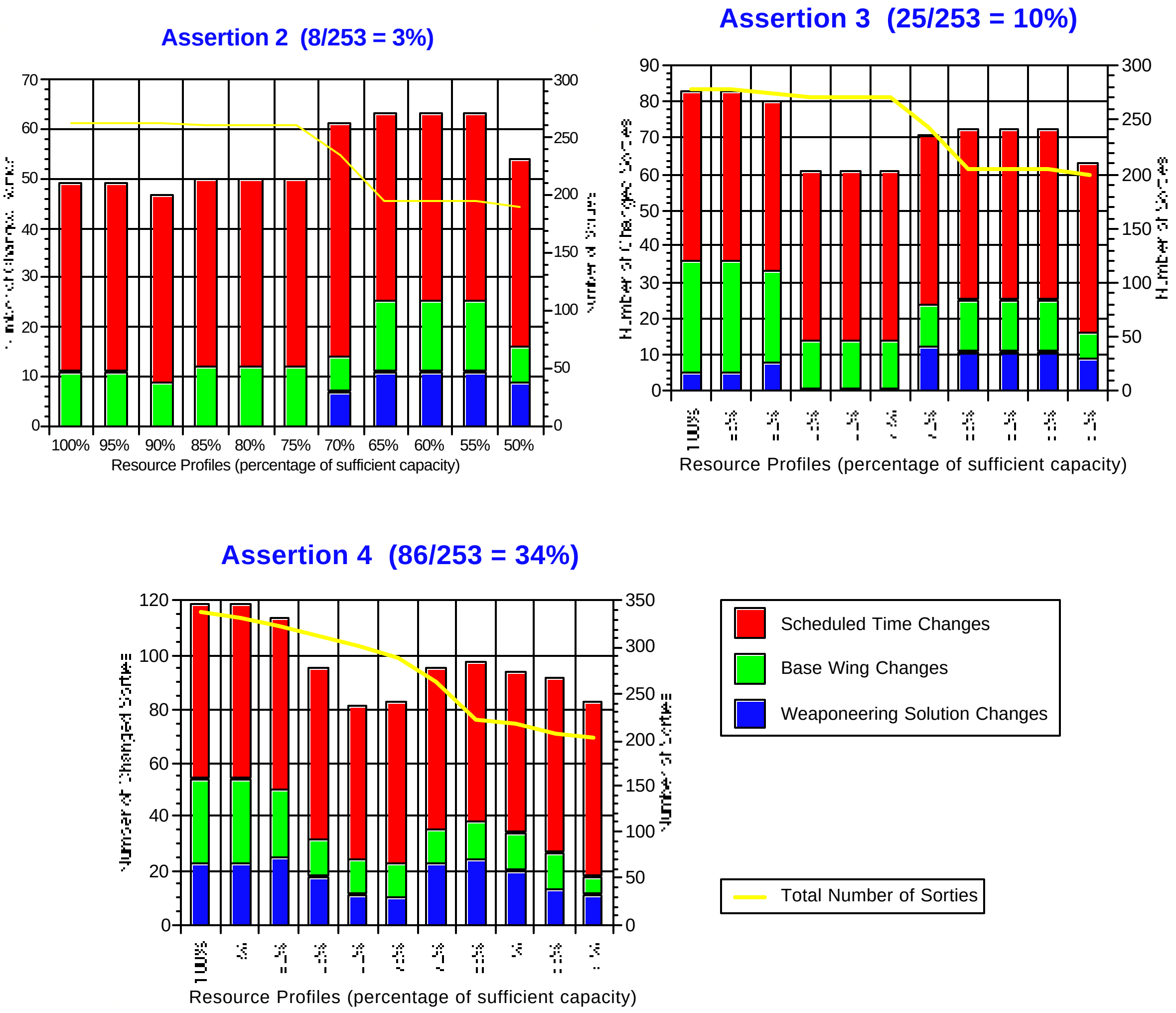

Figure 13. Changes to Initial Schedule – Regenerative, Largest DMPI First

The regenerative and incremental strategies tested above can in some sense be seen as extremes – that is, a strategy that resolves the problem completely (with the possibility of a completely different schedule) versus a strategy that does not change anything in the existing schedule. If it is assumed that individual tasks to be scheduled are not created equal but instead have associated priorities, then it is possible to define an incremental strategy that provides a less extreme (and possibly more useful) performance tradeoff. To explore this possibility, a third priority-based rescheduling strategy was defined and evaluated. In brief, the priority-based rescheduling strategy evaluated is defined as follows. New tasks are added into the current schedule incrementally as before, but in doing so any resource capacity that is currently allocated to tasks of lower priority than the one currently under consideration is considered as available for use by the current task (in which case lower-priority tasks will be preempted). The full procedure cascades, so that any preempted task will itself be rescheduled (if possible) and may preempt another still-lower-priority task in the process.

To evaluate this priority-based rescheduling scheme, priorities were randomly assigned to all tasks in the base plan (and hence all tasks in the current schedule). The new tasks associated with two of the new objectives, Assertion 2 and Assertion 3, were then assigned a higher priority than any task in the base plan/schedule (to ensure that they would indeed make it into the revised schedule). All other aspects of the original experimental design were left the same. We compared two sets of strategies: (1) prioritized, incremental rescheduling (as described above) and (2) prioritized regeneration (in which case the ordering heuristic is augmented to first consider priority and secondarily consider the base criterion).

Figure 14 presents the computational tradeoff. As might be expected, there is a crossover point as resource contention increases where it becomes more efficient to simply regenerate than to selectively disrupt lower-priority tasks. The position of the crossover point appears to depend on the size (or proportion) of the requested change.

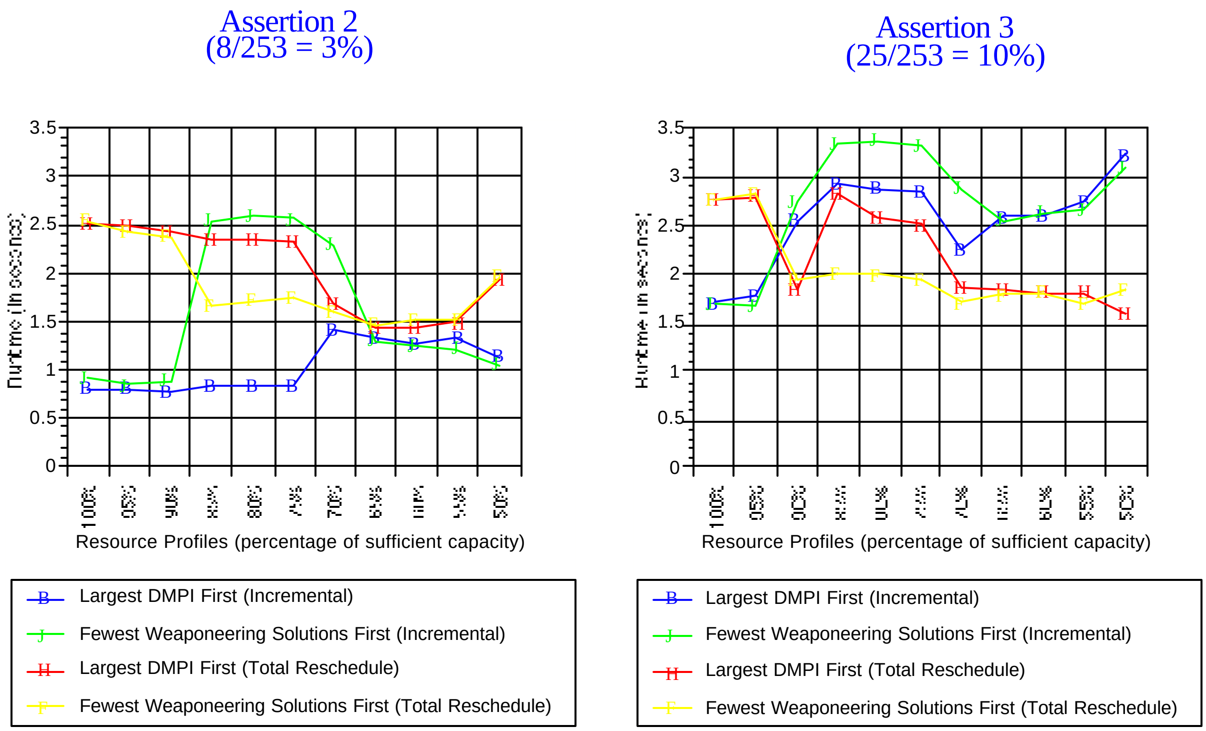

Figure 14. Computational Tradeoff – Priority-Based Scheduling

Figure 15 shows relative performance from the standpoint of solution quality. Again, no appreciable difference is seen between incremental and regenerative strategies.

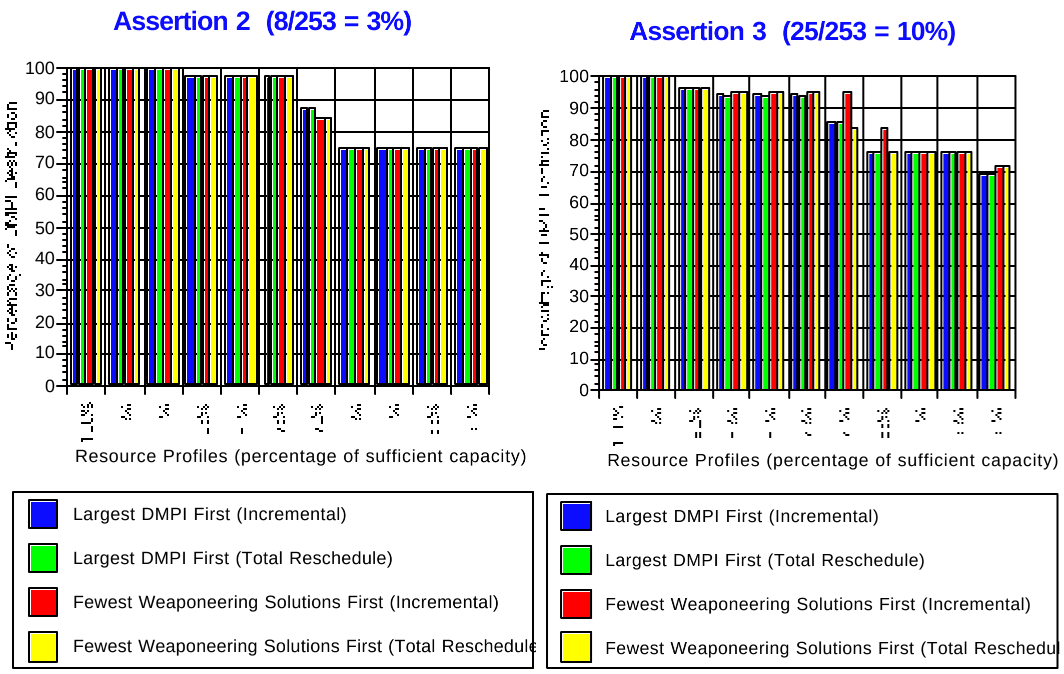

Figure 15. Solution Quality – Priority-Based Scheduling

Figure 16 shows the percentage increase in the extent of solution change resulting from regenerative instead of incremental priority-based scheduling. The results are averaged over runs for Assertions 2 and 3. It can be seen that the incremental strategy indeed provides a viable middle ground with respect to maintaining stability in the plan.

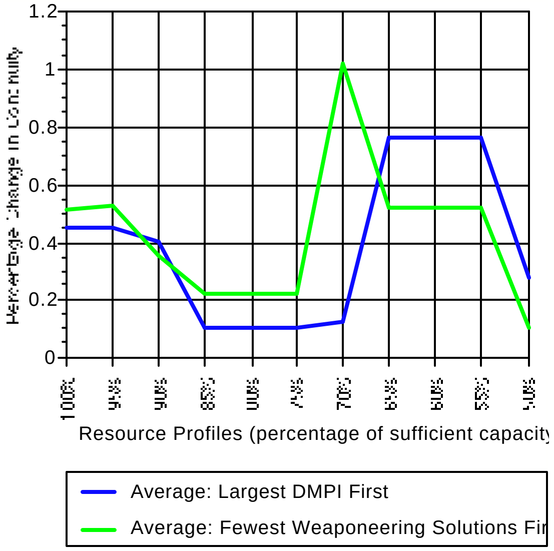

Figure 16. Percentage Increase in Solution Change – Regenerative versus Incremental Priority-Based Scheduling

Although the priority-based scheme tested above demonstrates a middle-ground with respect to balancing stability, solution quality and solution cost concerns, it is a heuristic strategy and it may be possible to improve it. One important aspect of the heuristic strategy is the approach taken to selecting which lower-priority tasks to "bump" when there are several alternative possibilities. The base scheme evaluated above uses an *Interval-based* approach to selecting a set of tasks to bump. In brief, the heuristic performs a "left-to-right" scan of a given resource profile over the interval where capacity is required, collecting lower-priority tasks based strictly on task priority and amount of capacity required. Analysis of this heuristic has led to the design of an improved *Task-based* approach. Under this task-based scheme, lower-priority tasks are selected from the resource profile in a more opportunistic fashion, based on a combination of priority and "best fit" from a required *resource area* perspective. The reader is referred to [Zhou and Smith 2001] for full technical details.

Figure 17, Figure 18, and Figure 19 show comparative results of these two pre-emption strategies (labeled "original" and "new") along computational, solution quality, and solution stability dimensions. The new task-based approach is seen to incur a greater computational cost and yield roughly the same quality solutions (in terms of number of DMPIs accomplished. However, with this slight increase in computation time (a few seconds on the largest problem tested), there is a significant, across the board improvement in solution stability with no appreciable difference in solution quality. It seems clear that Task-based approach gives rise to a stronger overall priority-based scheduling mechanism.

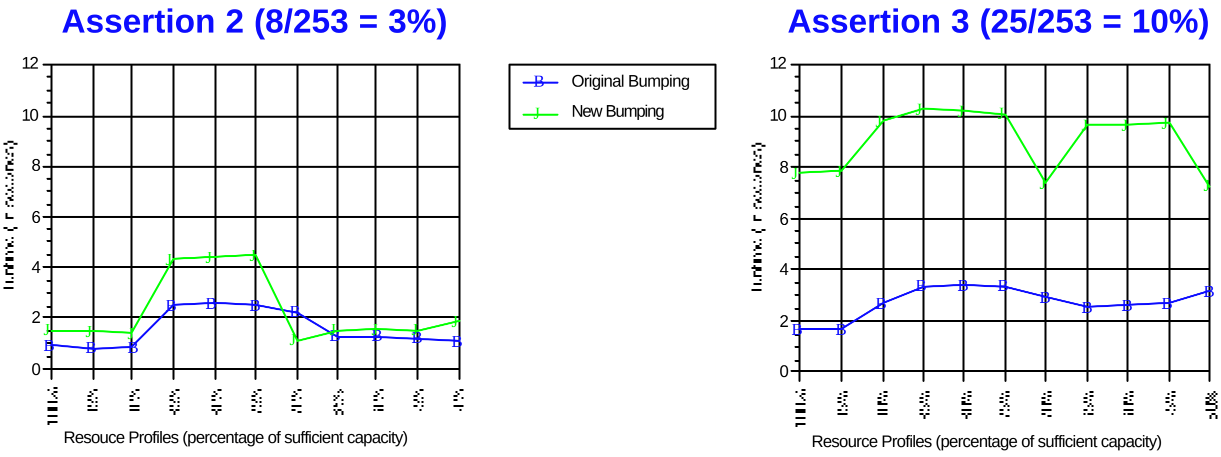

Figure 17: Computational Tradeoff between Interval-based andTask-based Pre -emption Strategies

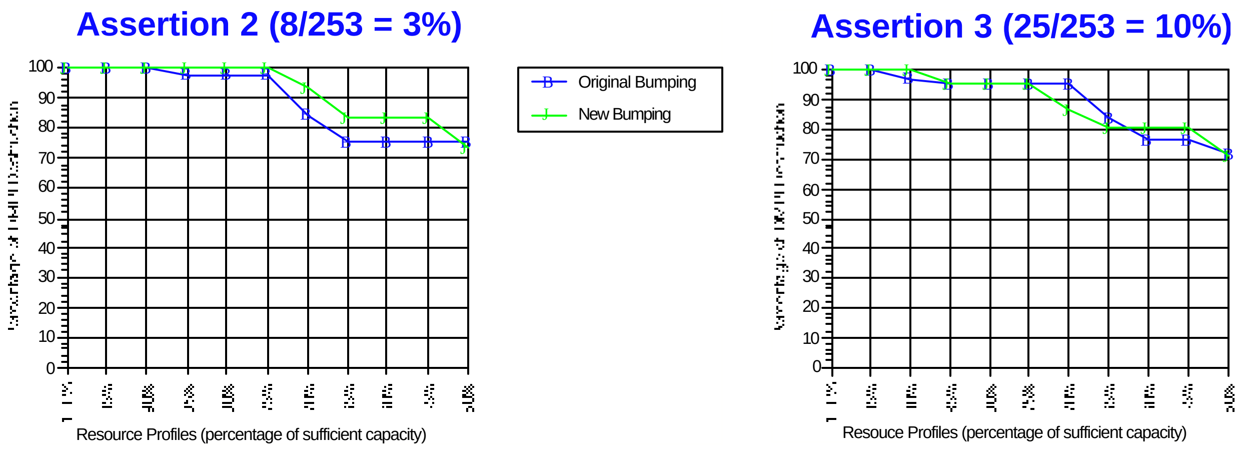

Figure 18: Solution Quality Tradeoff between Interval-Based and Task-Based Pre -emption Strategies

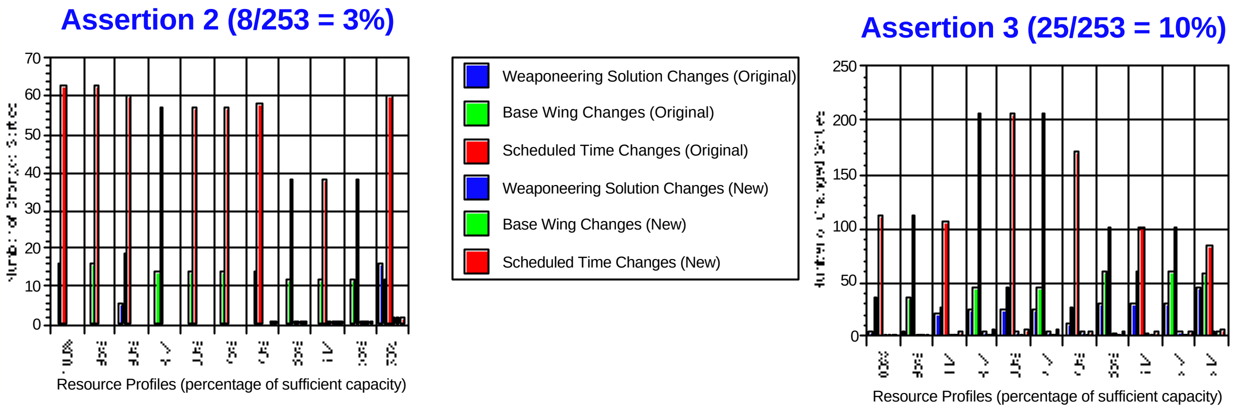

Figure 19: Stability Tradeoff between Interval-Based and Task-Based Pre -emption Strategies

6.4.2. Experiment B2: Response to Loss of Resource Capacity

The study described above considered a “top-down” response to the introduction of new objectives, and was concerned with understanding the reactive behavior of each component in this context. No attemp t was made to feed back any “unsupportable” missions that resulted from the plan/schedule modification process (although the strategy explored in Experiment A is directly relevant here as well).

In this section, we undertake a complementary analysis of interleaved planning and scheduling strategies for “bottom-up” response to such events as a reported loss in resource capacity. In this case, it may be possible to do some amount of reallocation so as to minimize impact on the plan itself. However, for large enough disruptions to resource availability, modification of the plan will be inevitable. In general, we assume the following interleaved process for response to resource status updates[1]:

[1] We restrict attention here to a reactive process aimed at cases where unexpected events have caused the planning/scheduling constraints to become tighter (and the solution is no longer feasible). Note however, that the

A. Invoke Schedule Repair Process – First, determine the set of affected (i.e., now unsupportable) tasks, and then reallocate resources as possible to (at least partially) absorb the disruptive event

B. Invoke Plan Repair Process – If there are remaining unsupportable tasks, modify the plan to reduce demand for over-subscribed resources

C. Invoke Schedule Repair Process – Revise the schedule to excise outdated tasks and integrate new plan fragments

D. Iterate steps B and C as necessary

Within the above process for reacting to unexpected events, the types and amount of information that is communicated between scheduler and planner orients the overall repair strategy and can be expected to have an impact on the effectiveness of the plan/schedule repair results. In the B2 experiment summarized below, we evaluate two specific repair strategies:

- *Intensity-Driven* – This approach, driven by the planner, determines an appropriate planning response by evaluating the current demand in the schedule, selecting an objective (or set of objectives) to reconsider in an attempt to help resolve any shortfall caused by the resource failure, and modifying the corresponding subplan(s).
- *Resource (Task) Driven* – This approach, driven by the scheduler, applies a set of heuristics to localize the impact of the resource failure and passes that affected set of tasks to the planner to suggest the area of the overall plan in which to replan.

Within either approach, various heuristics can be employed to localize the impact of a resource failure, depending on whether planning or scheduling concerns are considered. To favor an intensity-driven (i.e., planning) perspective, the scheduler can attempt to localize unscheduled tasks according to the number of objectives that will be affected or the size of any affected objectives (in terms of number of constituent tasks), in an attempt to minimize the required effort by the planner. From a resource or task driven (i.e., scheduling) perspective, the attempt can be made to limit schedule disruption by minimizing the number of unscheduled tasks and/or minimizing the number of time, base, and weaponeering-solution changes that would result from any rescheduling of the affected tasks. In the experiment performed, we coupled an “unscheduling” heuristic aimed at minimizing the number of objectives affected with the intensity-driven strategy, and a heuristic for minimizing the number of tasks unscheduled with the resource-driven strategy.

To evaluate these two strategies, a set of problems requiring planner/scheduler response to the loss of aircraft capacity assumed in the original plan was defined. Using the same 253 DMPI Cyberland plan used as a base in Section 6.4.1 as the starting point, five failure cases involving the loss of increasing amounts of aircraft capacity were defined. Each strategy was used to respond to all five repair problems.

Figure 20 and Figure 21 characterize the relative computational efficiency of the two strategies. Both the number of planner/scheduler iterations and the computation time required for successful repair can be seen to increase as the magnitude of the resource capacity loss is increased.

process could easily be generalized to cover circumstances where the constraints on the plan/schedule have unexpectedly loosened and there are now opportunities to improve the plan/schedule.

However, with the exception of the one boundary (minimal loss) case, the resource-driven strategy consistently outperforms the intensity-driven approach.

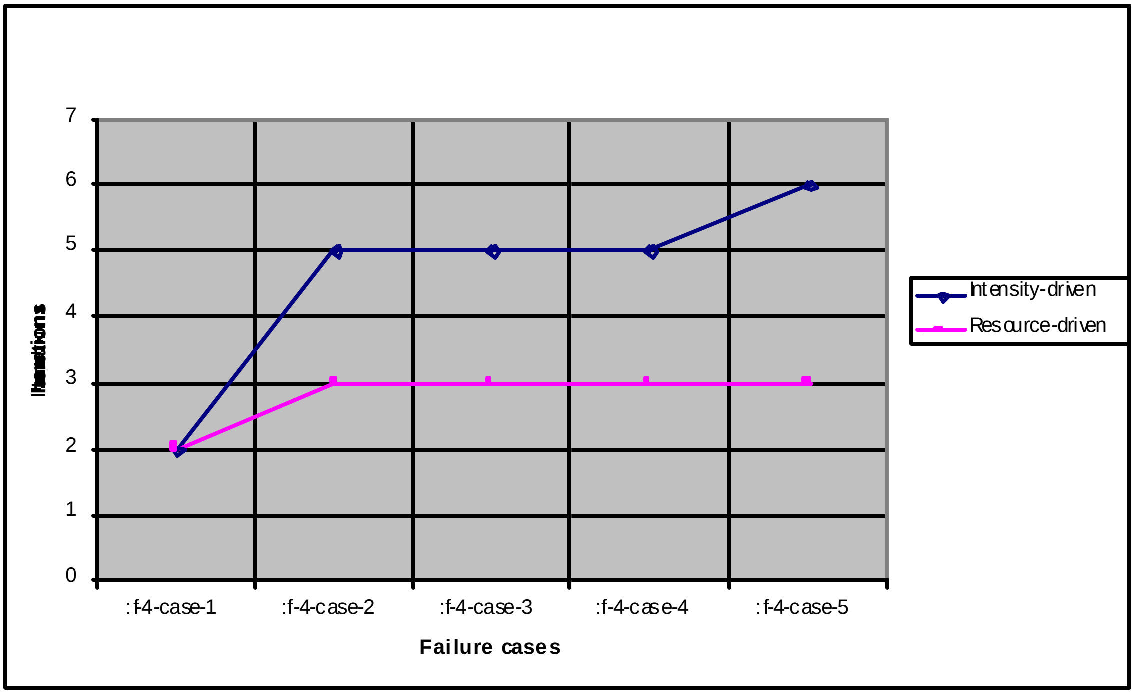

Figure 20: Number of Planner/Scheduler Iterations for Resource-Driven and Intensity-Driven Repair

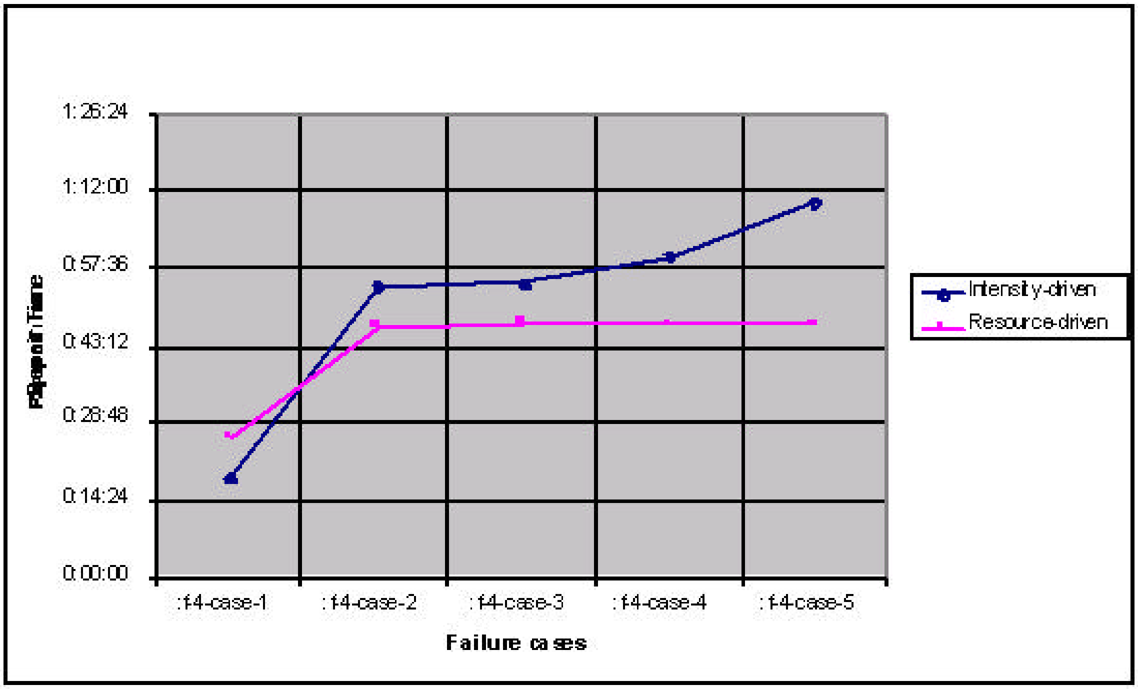

Figure 21: Time for Resource-Driven and Intensity-Driven Repair

Figure 22 shows the reduction in number of DMPIs that results from both resource-driven and intensity-driven repair strategies. In this respect, we see that the intensity-driven approach performs better, yielding an equivalent or smaller reduction in all cases.

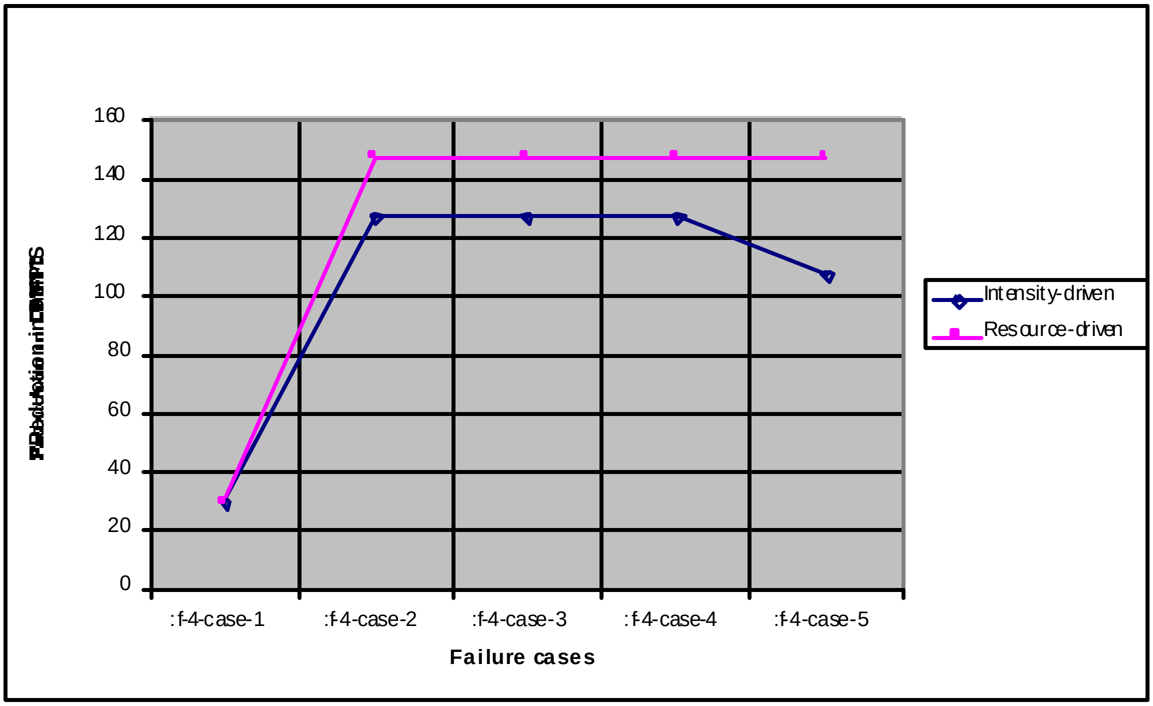

Figure 22: Reduction in DMPIs Covered for Resource-Driven and Intensity-Driven Repair

Finally, Figure 23 shows the comparative levels of change, from original schedule to final schedule, for both the intensity-driven and the resource-driven approaches. Here we see that the resource-driven approach produces a more stable (i.e., non-disruptive) repair with respect to the set of tasks common to both initial and final solutions.

6.4.3. Summary

Broadly speaking, the B1 Experiment confirmed our hypothesis with respect to the incremental integration of new objectives into an existing plan: *incremental strategies were generally found to be more efficient and less disruptive than regenerative strategies, without substantive degradation in solution quality.* The computational advantage of the incremental strategy was found to diminish as the level of resource contention and the relative size of the new plan addition increases. The crossover point is reached sooner in the case of priority-based solution change. With regard to solution stability, the expected continuum with respect to localization of change was confirmed. As a side result, our analysis of priority-based solution change led to the definition of a much more effective *task-based* strategy for pre-emption. The somewhat surprising result that incremental rescheduling was not seen to adversely affect solution quality is perhaps attributable to the greedy nature of the search that is conducted by the scheduler in both regenerative and incremental modes.

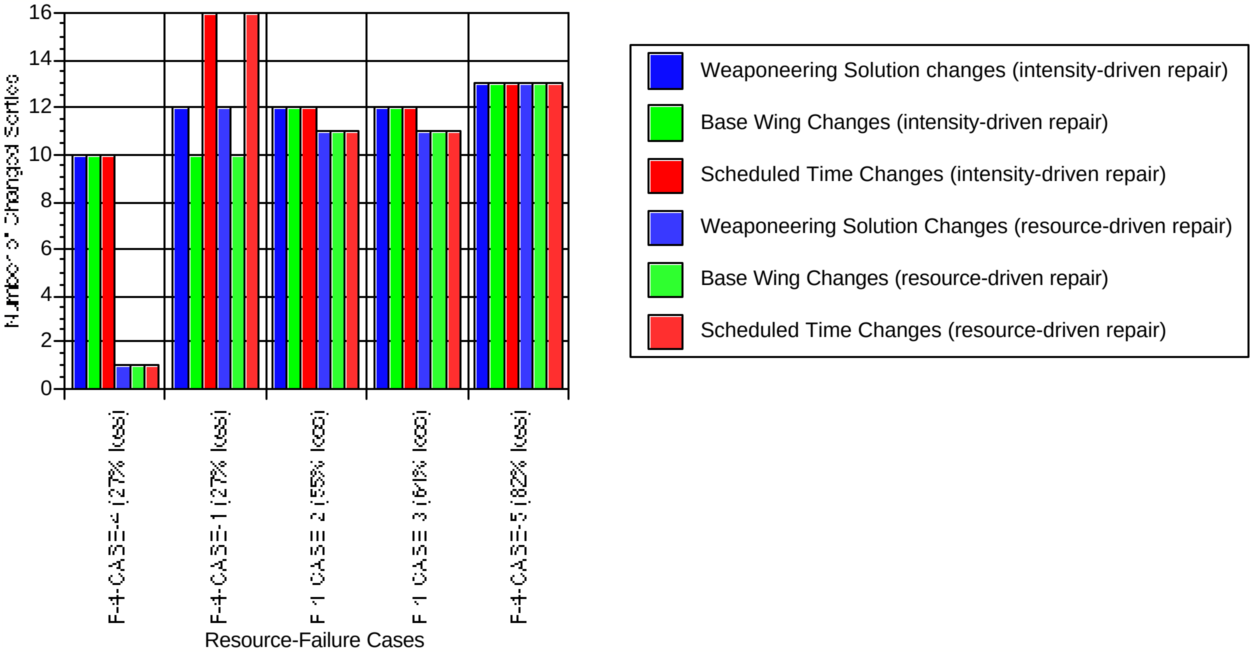

Figure 23: Schedule Stability for Resource-Driven and Intensity-Driven Repair

The B2 Experiment showed generally that a scheduler-driven repair strategy focused on tasks that have been found difficult to schedule leads to more streamlined interaction and a more efficient overall repair process than a planner-driven strategy that seeks to reduce resource intensity for subplans with resource contention. The intensity-driven approach, alternatively, was seen to preserve greater portions of the original plan and generally to get more accomplished with available resources. Finally, as might be expected, the scheduler-driven approach was found to cause less disruption in the schedules of those tasks common to both the initial and final plans.

The relative ineffectiveness of the planner-driven approach to plan/schedule repair was not expected. These results suggest that the single-dimensional approach to modeling the resource intensity of various planning alternatives may not be sophisticated enough to provide adequate guidance for repair.[2] The next section considers the design and evaluation of a more refined approach to representing and reasoning with resource intensity information.

[2] The B2 Experiment considered only a small set of failure cases tied to specific types of resource unavailability. As such, these results may not reflect general behavior.

7. Experiment C: Multidimensional Quantitative Intensity Model

This section describes a third integration experiment designed and performed to evaluate the advantages of a multidimensional approach to modeling resource intensity. The description provided below conforms to the documentation format provided by the JFACC program.

7.1. Statement of Experimental Objectives

7.1.1. Hypothesis

Tightening the representational coupling between a planner and scheduler will increase further the benefits of considering resource allocation issues during planning.

Level: Process Experiment

Description:

Experiment A confirmed the hypothesis that the early consideration of resource allocation issues during planning can improve efficiency substantially. This experiment extends those investigations by exploring a richer representation for supporting cooperation between the planner and scheduler during the plan generation and repair processes.

The earlier experimentation focused on a simple *qualitative* model of expected resource usage grounded in a single aggregated *intensity* number. In contrast, Experiment C explores a *multidimensional quantitative* approach that models expected resource usage at a finer level of granularity. In particular, resources are grouped into functional categories intended to capture similarities in how resources are applied. These groupings provide an aggregation over the individual resource classes, thus simplifying the resource models inherent to the scheduler; however, the aggregation has greater detail than the individual number afforded by the single-dimensional intensity model and so would be expected to have improved predictive value for estimating resource usage.

The main experiments in this set will compare the time required to generate a scheduled plan by the single-dimensional, multidimensional, and waterfall approaches. The multidimensional approach incorporates several control parameters that can be adjusted to tune its behavior; additional experiments will evaluate the sensitivity of the approach to each.

7.1.2. Value

This experiment set enhances the benefits to command and control outlined for Experiment A in Section 5.1.2. The higher-fidelity approach to resource summation will reduce further the time taken to produce quality plans and schedules. This improved capability will in turn enable commanders to explore an even greater number of courses of action.

7.2. Description of Experimental Setup

7.2.1. Simulation Features

This experiment did not require a simulation environment and so was conducted independently of the Enterprise Models.

7.2.2. Variables or Correlated Parameters

Major independent variables: resource availability. The multidimensional technique relies on a number of control parameters; sensitivity of the approach to each will also be evaluated.

7.2.3. Specification of Test Runs

The improvements yielded by the higher-fidelity coupling will depend on the degree of resource constrainedness of the underlying problem: greater benefits should result for problems where resources of a particular category run short. The experiment set takes this factor into account by considering resource availability profiles that provide qualitatively distinct points along the spectrum of resource constrainedness. For several points along this spectrum, we define a distribution of resource profiles and compare both the single and multidimensional approaches.

System performance is measured in terms of the computation times required for plan/schedule generation across various test runs, the number of planner/scheduler interactions required, and the quality of the resulting scheduled plans.

7.3. Results

7.3.1. Overview of the Multidimensional Intensity Approach

To provide the necessary context for the subexperiments that constitute Experiment C, we begin by describing more fully the multidimensional method.

7.3.1.1. Multidimensional Intensity Model

The multidimensional approach employs the same basic strategy for intensity adaptation in response to scheduler feedback that was described in Section 5.3. The main difference between the single-dimensional and multidimensional methods lies with the model of actual and expected resource usage that underpins the communication between the planner and scheduler.

In the single-dimensional approach, planning strategies are annotated with a single intensity value that reflects a qualitative indication of their relative resource usage in comparison to other strategies for the same task. The weakness of that approach lies with its lack of granularity. To see why, consider a situation with relatively low overall resource demand, but where the class of resources required to hit a specific type of target is nearly exhausted. The single-dimensional case would not be able to adjust strategy selection to adapt to the shortage because there is an overall abundance of resources. In contrast, because the multidimensional model can represent a lack of capacity for specified groups of resources, strategy selection could be adapted to prefer approaches with minimal demand for the oversubscribed resource.

Table 1 summarizes the four intensity dimensions employed in the experiments. Each dimension is defined by a group of munition types and the aircraft types that can carry those munitions. The *unguided* category corresponds to “dumb” or “gravity-guided” munitions that can be deployed from any of the available aircraft. *Precision* munitions require sophisticated guidance machinery on the delivery aircraft, a constraint that in turn limits the types of aircraft that can deploy them. *Cluster* munitions are chiefly antipersonnel and antiarmor munitions that can be carried by a specialized subset of the available aircraft. *Defense* munitions are used in SEAD missions and can be deployed only by aircraft fitted with sophisticated radar detection equipment.

Intensity Dimension	**Aircraft Type**	**Munitions**
Unguided	F-1, F-2, F-3, F-4, H-1A	MK1, AGM1
Precision	F-3, F-4, H-1A	MK2, AGM4
Cluster	F-3, F-4, F-2, F-1	Cluster1, Cluster2
Defense	F-4	AGM2

Table 1. Air Operations Intensity Dimensions

As can be seen from Table 1, the dimensions are not mutually exclusive. For example, F-3s appear in all but the *Defense* dimension, reflecting the fact that those aircraft can be used to fly different types of mission. This connectivity introduces additional complexity into the multidimensional intensity adaptation process, since decisions related to one dimension can impact results for other dimensions.

7.3.1.2. Resource Capacity Model

Resource availability and usage is measured in terms of *capacity*. As defined in Figure 26, capacity for a given aircraft type *A* is defined to be the product of the number of available aircraft of type *A*, the *DMPI-Rate* for *A*, and the overall duration of the operation.

$$Capacity(A) = Count(A) \leftrightarrow DMPI - Rate(A) \leftrightarrow OpDuration$$

Count(A): number of available aircraft of a given type

DMPI-Rate(A): DMPI prosecution rate for a given aircraft type

OpDuration: duration of operation

Figure 24. Aircraft Capacity Determination

The definition of capacity for a particular intensity dimension is grounded in the capacity for each aircraft type within the dimension. Those capacities must be adjusted, however, both to reflect the overlap in aircraft type among intensity dimensions (as noted above) and to reflect the expected relative contribution of each dimension to the overall plan size. For example, if one dimension is expected to yield two-thirds of the resource requirements within a plan, its capacity should be adjusted to reflect that bias. To this end, we employ *allotment factors* to define the relative proportion of resources that should be accorded to each dimension. Table 2 summarizes the baseline allotment weights used within our experiments.

Intensity Dimension	Allotment Weights
Unguided	.35
Precision	.05
Cluster	.35
Defense	.25

Table 2. Baseline Allotment of Resources to Dimensions

It should be noted that the allotment weights need only be rough approximations; furthermore, it is reasonable to expect that a commander would have a good sense of the relative distribution of resources among the different target types. Experiment C5 investigates the sensitivity of the multidimensional intensity adaptation method to the allotment values.

The capacity for an intensity dimension is defined to be the sum of the capacity for each aircraft type in the dimension, scaled in accord with the allotment factor that weights resource distribution for the dimension:

$$Capacity(I_j) = \sum_{A \in AircraftType(I_j)} Allocation(I_j) \times Capacity(A)$$

7.3.1.3. Multidimensional Adaptation Method

For the multidimensional intensity approach, the incorporation of resource information to guide the selection of planning strategies involves three steps. First, for a given subplan, the *target intensity* for each dimension must be determined, based on the resources available and the number of subplans that remain to be generated. Second, for a given planning decision, each applicable planning strategy must be rated according to its resource requirements in each intensity dimension. Third, with each available strategy labeled with its resource intensity and the ideal intensity given current resource usage calculated, the closest planning strategy to the ideal is determined. Each stage in this process is described in detail below.

Target Intensity

The target intensity, denoted by I^T, represents the expected 'ideal' use of resources for a particular subplan, relative to current availability and expected demand. Calculation of the multidimensional target intensity specializes that of the single-dimensional approach (presented in Section 5.3) to each dimension. The intensity for a given dimension is defined in terms of the ratio of the resources available per remaining subplan to the resources allotted originally to each subplan; this ratio is then normalized relative to the interval of intensity values in use (namely, *[0, TopIntensity]*).

Let *Capacity(Ij)* be the overall capacity for resources in dimension *j* and let R_j^i be the remaining capacity for dimension *j* after the first *i* of *n* subplans have been created and scheduled. The following equation defines the target intensity I^T for the *i+1st* subplan:

$$I^T = \begin{bmatrix} I_1^T \\ \vdots \\ I_m^T \end{bmatrix} \quad where \; I_j^T = \frac{\frac{1}{n-i} \times R_j^i}{\frac{1}{n} \times Capacity(I_j)} \times TopIntensty$$

Provided that resource usage remains below allotment levels, the value of I_j^T will exceed *TopIntensity.* Values below *TopIntensity* indicate that planning choices should seek to decrease demand for resources within that dimension below the original allotment level.

Operator Intensity

Figure 25 presents the equations for the intensity of a planning operator O_k , which is denoted by I^{Oi}. The intensity for each dimension is defined to be the ratio of the expected resource demands introduced by the operator to the original allotment of resources for that subplan and dimension (assuming uniform allotment among subplans).

$$I^{O_k} = \begin{bmatrix} I_1^{O_{ki}} \\ \vdots \\ I_m^{O_{ki}} \end{bmatrix}, \text{ where } \quad I_j^{O_{ki}} = \frac{ExpectedDemand(O_k, I_j)}{\frac{1}{n} \times Capacity(I_j)} \times TopIntensity$$

Figure 25. Multidimensional Operator Intensity

For the air operations domain, the resource demands of an operator are measured in terms of the expected munitions and aircraft required to prosecute the targets associated with the operator. These estimates are calculated by summing the expected number of DMPIs of a given type multiplied by a capacity estimate for the type. For example, the strategy of attacking the maneuver and armor capabilities of a ground force may have an estimate of one DMPI per tank and three DMPIs per bridge.

Operator Ranking

The first two steps of the selection process yield a vector of target intensity values (*I^T)* together with a set of applicable planning strategies, each labeled with a vector of required intensity values (I^{Ok}). Each applicable operator is assigned a rating that scores how closely the operator's intensity requirements match the target intensity values. Operator selection then reduces to identifying the operator with the best rating.

Figure 26 presents our scheme for ranking operators according to their proximity to the target intensity values. The ranking method builds on the *intensity difference vector* $D^{Ok}= I^T\text{-}I^{Ok}$, which gives the difference between the target intensity and operator intensity vectors. The operator rating, denoted by *Rating(O^i),* is defined to be the sum of the magnitudes in the intensity difference vector, adjusted by a *penalty factor*.

In cases where the difference value d_j is positive (i.e., the operator requires fewer resources than indicated by the target intensity), the penalty is defined by P^+; in cases where $d_j<0$ (i.e., the operator is expected to use more resources than indicated by the target intensity), the penalty is defined by P^-. Through appropriate settings of the ratio of these penalty factors, different strategies can be defined that penalize resource overutilization/underutilization to different degrees.

$$Rating(O^k) = \sum_{d_j \in D^{O_k}} Penalty(d_j)$$

$$Penalty(d) = \begin{cases} P^+ \times d & for\ d \geq 0 \\ P^- \times ABS(d) & for\ d < 0 \end{cases}$$

Figure 26. Operator Ranking

With this rating scheme, the operator with the lowest rating will be preferred.

7.3.2. Experiment Setup

7.3.2.1. Test Domain

Experiments A and B built on a domain model for establishing air superiority that SRI developed prior to the start of this phase of the JFACC program. The main objective in developing that domain was to provide a high-fidelity characterization of a key component of air operations in order to demonstrate the suitability of our automated planning tools for this type of application. In particular, support for experimentation was not a consideration in the development of the air superiority domain.

For Experiment C, we developed a complementary domain that covers ground-force interdiction missions. Two goals motivated the construction of this new domain. First, it was designed explicitly to support experimentation, while also maintaining reasonable fidelity to the operational problem. In particular, the complexity of the domain can be more readily controlled, enabling the experiments to uncover the sensitivity of a planner method to subtle changes in problem structure. Second, the new domain provides the opportunity to demonstrate the value of our intensity-based planner/scheduler methods for different types of problem.

The ground-interdiction domain employs the same basic target network model that underlies the original air superiority domain. Objectives within this domain reduce to the task of neutralizing enemy capabilities organized as networks of targets. Here, each network corresponds to an enemy ground-force unit with maneuver, personnel, armor, and artillery components. We provide several strategies for attacking networks: attacking all components, attacking a coherent subset of capabilities (e.g., heavy weapons – armor and artillery), or attacking just a single capability.

Our primary test problem for this domain involves interdiction of ground-force units. The problem yields plans with 8 subplans and from 50 to 724, actions depending upon the aggressiveness of the planning strategies applied. As part of our experimentation process, however, we consider variants of this main problem that involve different numbers and structures of forces, resulting in a broad variety of plans and schedules.

This new problem domain required the definition of a corresponding set of resource profiles to match the scope and types of operation produced by the planner. Figure 27 summarizes the set of resource profiles used throughout Experiment C. For simplicity, we varied aircraft availability only; sufficiently large numbers of munitions were made available to avoid any munition-based resource conflict. The maximum or 100% profile provides just sufficient resources to support the maximum plan; the profiles then decay gradually until there are insufficient resources to support the minimal plan. In addition, the experiments employ a profile called ‘Big’ that contains a large amount of all aircraft resources relative to the maximal plan.

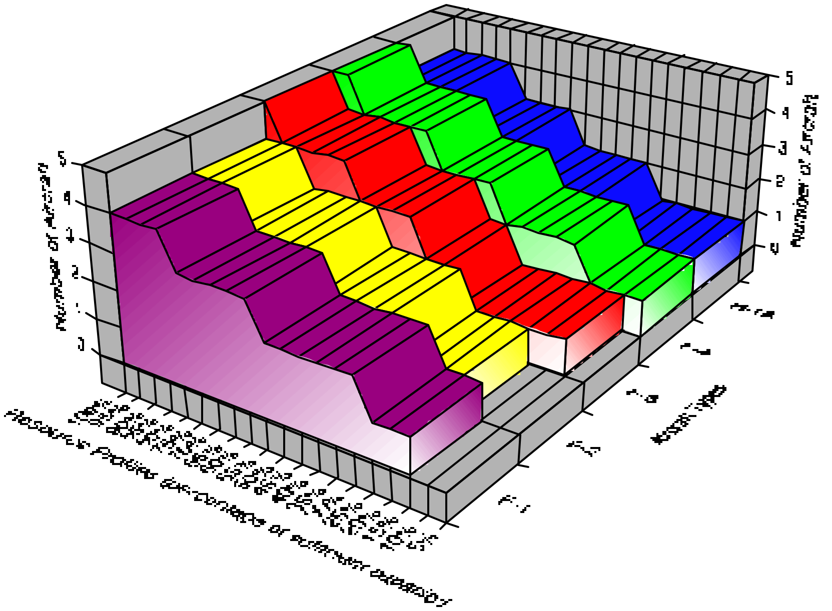

Figure 27. Experiment C Resource Profiles

7.3.3. Overview of experiments

Experiment C consists of five related experiments designed to evaluate the multidimensional intensity adaptation method.

The main experiment (C1) examines the performance of the multidimensional approach relative to the single-dimensional and waterfall methods on the primary ground-interdiction problem. For performance, we consider three main factors: generation time, plan size, and number of planner/scheduler interactions. As noted above, we use plan size as a ‘stand-in’ for evaluation plan quality; this approach derives from the philosophy that, for a given allotment of resources, a given commander will seek to maximize the aggressiveness of his plan as a way of increasing the likelihood of satisfying his objectives.

To ensure the generality of our results, we performed two experiments that vary the basic problem under investigation. One variation involves changing the distribution of sizes of ground-force networks, resulting in subplans with differing structures and numbers of actions (C2). The second variation considers changes in the degree of interdependency among ground-force networks (C3).

Our final experiments in set C evaluate the sensitivity of the multidimensional method to its two main classes of parameters, namely *resource overutilization/underutilization penalties* (C4) and *resource allotment weights* (C5).

7.3.4. Experiment C1: Comparison of Waterfall, Single -dimensional, and Multidimensional approaches

Experiment C1 tests the hypothesis that the multidimensional approach will require less generation time than the single-dimensional method, without significantly impacting solution quality. Comparisons with the waterfall approach are included to provide further insights into the benefits of resource feasibility checking during planning.

Figure 28 shows the results for Experiment C1. The upper-left graph displays generation times in log scale for the three methods; the upper-right graph shows linear times for the single-dimensional and multidimensional cases. As can be seen, the waterfall method requires substantially more time once resources become constrained, while the intensity-based methods perform much better. As the linear-scale graph clearly shows, the multidimensional approach outperforms the single-dimensional approach, with the advantage increasing as resource availability drops. The bottom-left graph displays the number of interactions between the planner and scheduler required to find a solution. As with generation time, these results show that the multidimensional method outperforms the single-dimensional method, and that they both are far superior to the waterfall method as resources become more limited. Together, these results confirm the first part of the hypothesis for Experiment C, namely that the multidimensional approach would be faster.

These results show an impressive speedup by the intensity adaptation methods over the waterfall baseline. The tests used a scaled-down version of the domain in which goals that did not involve intensity decisions were limited to a single applicable operator. As such, the waterfall backtracking was limited to the same choices as the intensity adaptation methods. A side experiment was run where nonintensity goals had two applicable operators rather than just one. Runtimes for the intensity methods were virtually identical to those in Figure 28, since the intensity method backtracks at the level of intensity values rather than operators (hence, it is not impacted by the additional operators). In contrast, the waterfall method was unable to find a solution below the 100% resource profile after 239 trials and almost 30 hours of runtime. The waterfall method fails so badly in this larger problem because many planning decisions must be backtracked through to reach one that impacts resource usage significantly.

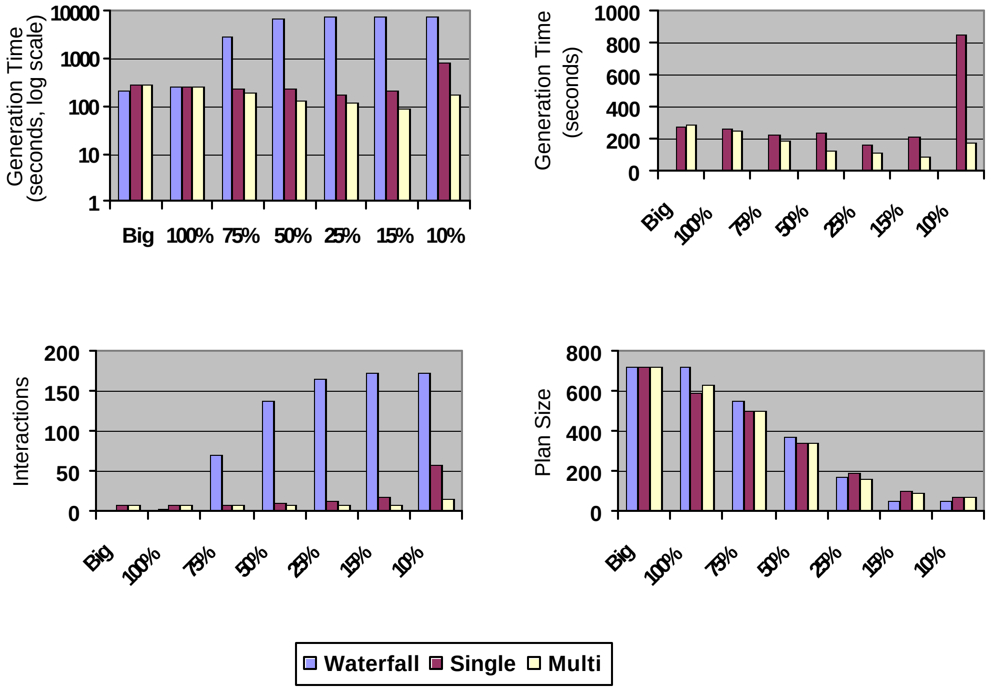

Figure 28. C1 Results: Comparison of Waterfall, Single-dimensional and Multidimensional Method

The waterfall approach produces slightly larger plans than the intensity-based methods for the 100% through 50% profiles; as resource availability decreases further though, it produces smaller (i.e., less aggressive) plans than the intensity-based methods.[3] In comparing runtimes, it is clear that the small increases in plan size for the waterfall method come at the cost of an increase of several orders of magnitude in planning/scheduling time. While there is some variation between the single-dimensional and multidimensional methods in plan size, the difference is relatively small. Overall, these results confirm the second part of the hypothesis of Experiment C: the performance benefits realized by the multidimensional approach do not adversely impact solution quality.

7.3.5. Experiment C2: Sensitivity to Subplan Structure

The problem used in Experiment C1 considered a set of ground-force networks of equivalent size and composition. In resource-rich cases (e.g., the Big profile), the generated subplans would

[3] As explained in Section 5.3, the waterfall method will not always find the largest plan.

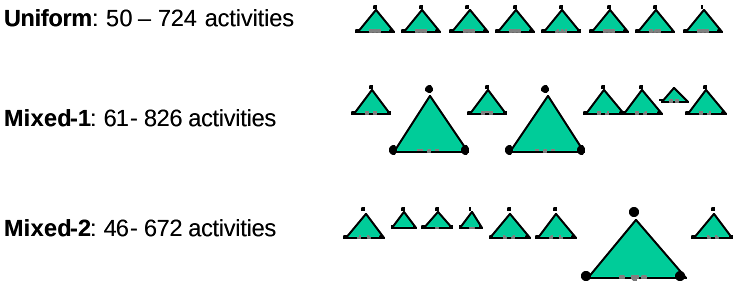
Uniform: 50 – 724 activities
Mixed-1: 61- 826 activities
Mixed-2: 46- 672 activities

Figure 29. Problem Topologies for C2 Experiments

have uniform structure and size. In more resource-constrained runs, however, the need to react to resource shortfalls would yield variations in subplan structure and size.

To test the sensitivity of the multidimensional approach to subplan structure, we ran the single-dimensional and multidimensional tests described in Experiment C1 on two additional problems. The topologies for the plans that they produce for the ‘Big’ profile are shown in

Figure 29. “Uniform” corresponds to the problem from Experiment C1. “Mixed-1” contains two large and one small subplan together with five of the same size as those used in the “Uniform” case. The “Mixed-2” problem contains three small, one large, and four of the same size as those used in the “Uniform” case, with the large subplan appearing late in the plan.

The tables in Figure 30 summarize the generation time and plan size for all three problem topologies; Figure 31 and Figure 32 provide graphical displays of the generation time, plan size, and number of interactions for the Mixed-1 and Mixed-2 problems, respectively. In Figure 30, entries marked by “*”indicate that the case ran for more than 24 hours without finding a solution; bold entries mark the best result for a given problem. The Mixed-1 entry for the 10% resource profile is greyed out in all cases; that profile lacks sufficient resources to solve the Mixed-1 problem within the operational time window imposed for the planning/scheduling problem.

The nonuniform problems proved to be a bit more difficult for the multidimensional approach to solve, resulting in slightly more planner/scheduler interactions and higher generation time relative to the uniform case. In contrast, the single-dimensional approach shows much more sensitivity to nonuniform subplan topologies for generation time, especially at the lower resource profile levels. The variations in topology have only minimal impact on plan size for the intensity methods. The waterfall method fared much worse than the intensity methods on both of the nonuniform problems. In particular, the waterfall method was unable to find solutions to the Mixed-1 problem for any but the Big resource profile. The lack of success for the waterfall method on the Mixed-1 stems from the need to backtrack a long way to reach the subplans with significant resource usage.

In summary, the C2 results confirm the general conclusion from Experiment C1 that the intensity-based methods far outperform the waterfall method without degrading solution quality.

Generation Time

		Big	100%	75%	50%	25%	15%	10%
Uniform	Multi	291	**254**	**196**	**134**	**120**	**90**	**175**
	Single	277	268	226	242	171	220	858
	Waterfall	**213**	266	2827	6844	8061	8223	8223
Mixed-1	Multi	327	256	**204**	**138**	**107**	**187**	
	Single	324	**235**	231	191	222	2066	
	Waterfall	**247**	*	*	*	*	*	
Mixed-2	Multi	278	255	**281**	**172**	**124**	**89**	**335**
	Single	264	277	297	283	499	336	5821
	Waterfall	**222**	**222**	4194	15474	16814	16991	16991

Plan Size

		Big	100%	75%	50%	25%	15%	10%
Uniform	Multi	**724**	636	504	348	164	90	**72**
	Single	**724**	592	504	348	**190**	**106**	**72**
	Waterfall	**724**	**721**	**554**	**368**	174	54	54
Mixed-1	Multi	**823**	**629**	**497**	**331**	**187**	**105**	
	Single	**823**	603	471	313	183	**105**	
	Waterfall	**823**	*	*	*	*	*	
Mixed-2	Multi	**677**	589	465	**355**	160	104	**132**
	Single	**677**	**612**	501	323	**179**	**106**	72
	Waterfall	**677**	**612**	501	323	**179**	**106**	72

Figure 30. C2 Results: Uniform, Mixed-1, and Mixed-2 Topologies

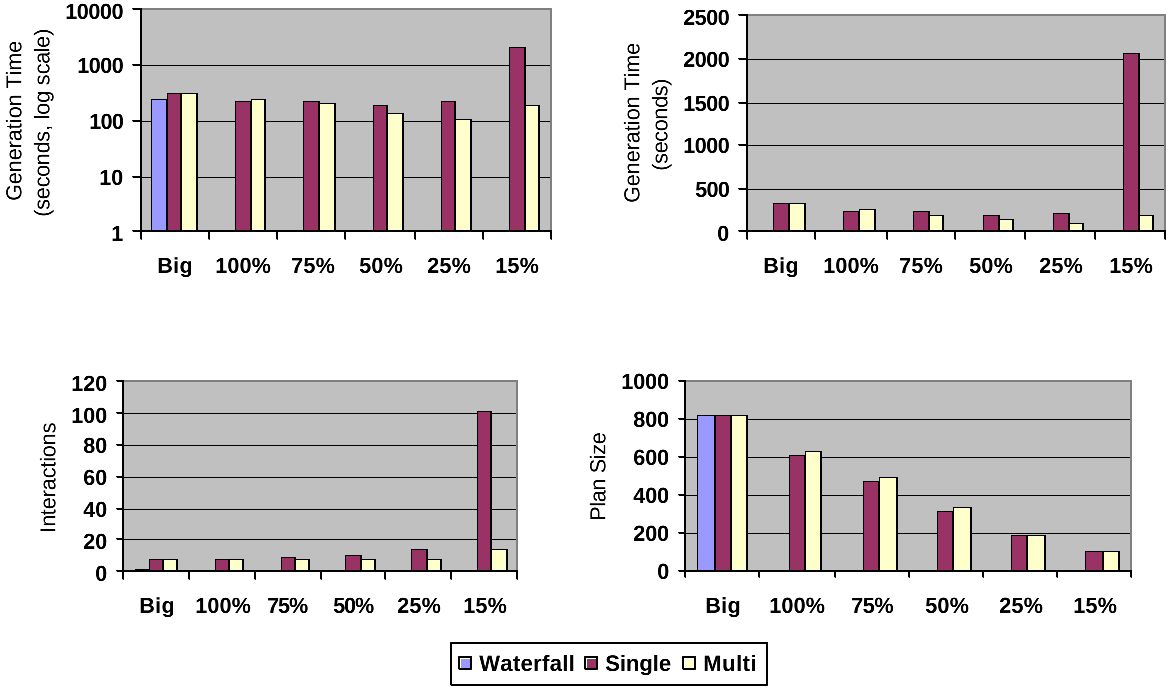

Figure 31. C2 Results: Problem Mixed-1

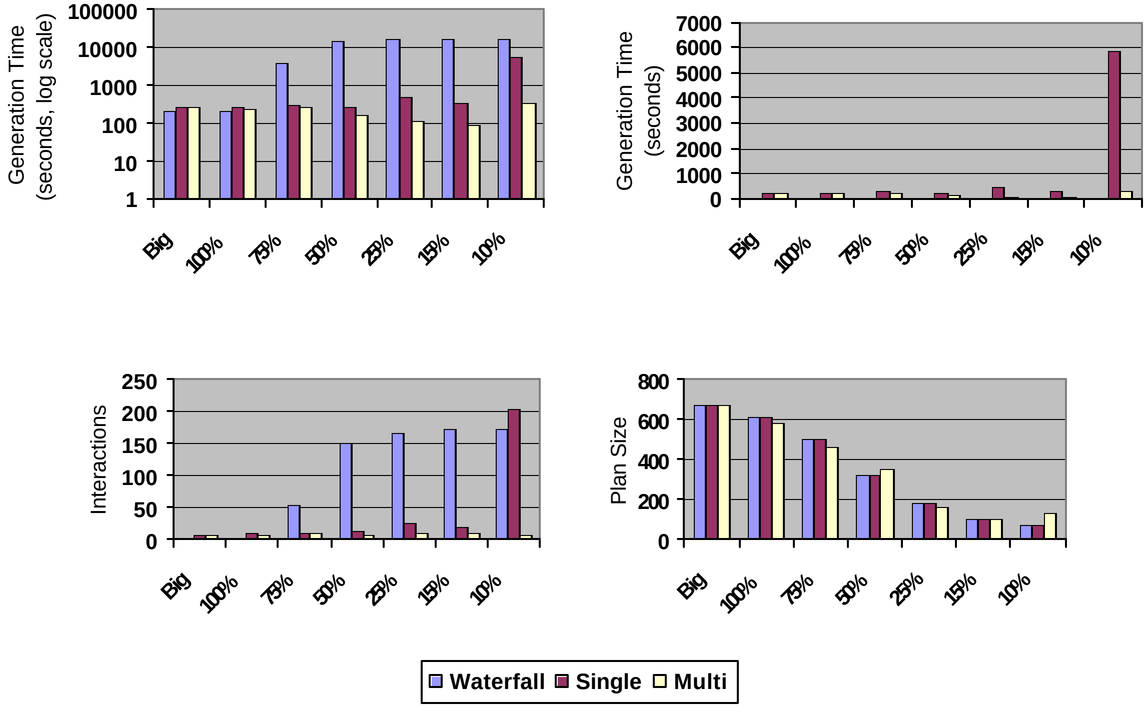

Figure 32. C2 Results: Problem Mixed-2

7.3.6. Experiment C3: Sensitivity to Subplan Interdependencies

The test problems used in Experiments C1 and C2 lacked any strategic dependencies among subplans at the planning level. In particular, the only interdependencies among these subplans arose at the scheduler level as a result of resource contention. Experiment C3 focuses on evaluating the sensitivity of the intensity adaptation methods to interdependencies among subplans.

The experiment emphasizes one specific type of interdependency, namely *goal phantomization*. Phantomizations occur when actions in one part of a plan have the beneficial side effect of achieving objectives in other part(s). For example, a plan may contain one objective to prevent red ground forces from reaching a particular border sector and second to prevent red attacks on infiltrating blue forces in a nearby area. In such a situation, it could be that destroying a certain bridge will facilitate achievement of both objectives. Thus, an action in the subplan for the first objective would have the beneficial effect of helping to achieve an objective in a different subplan.

To enable experimentation with phantomized goals, the domain model was modified to include SAM sites that protect multiple targets. Degrading a SAM as part of attacking one target would then have the side effect of removing the air defense from other targets defended by that same SAM. To focus on the impact of phantomization, only the resources needed to attack SAMs were constrained; sufficient munitions were provided to attack all ground targets. An inability to identify a potentially phantomized SAM at intensity computation time can lead to the mistaken assumption that additional resources will be needed to neutralize the SAM; if SAM striking resources are running low, the planner may conservatively choose to avoid that target by selecting a less-intense planning strategy.

Four problems were defined, each with increasing degrees of phantomization. The number of SAMs that protect two targets varied through 0, 6, 12, and 18; all other SAMs protect only one target. The range of resource profiles considered provides one, two, and five SAM-striking aircraft (F4s). Figure 33 presents the generation time, planner/scheduler interactions, and plan size results for all combinations of the four problems and three resource profiles. The labels for the x-axes embed the number of F4s in the resource profile and the number of SAM overlaps in the test problem.

For the cases involving five and two F4s, variations between the single- and multidimensional cases are relatively small. When there is only a single F4 available, however, substantial differences arise. Here, the single-dimensional approach takes substantially more time and requires many more interactions than does the multidimensional approach. In this situation, the multidimensional approach is able to determine early on that there is a shortfall of a critical resource (i.e., F4s), and it adjusts intensity accordingly to avoid overusing that resource. In contrast, the single-dimensional approach cannot distinguish F4s from other resources and so attempts to be much more aggressive in its planning strategy. This causes the significant increase in generation time and interactions; it does, however, have the beneficial side-effect of yielding somewhat larger plans.

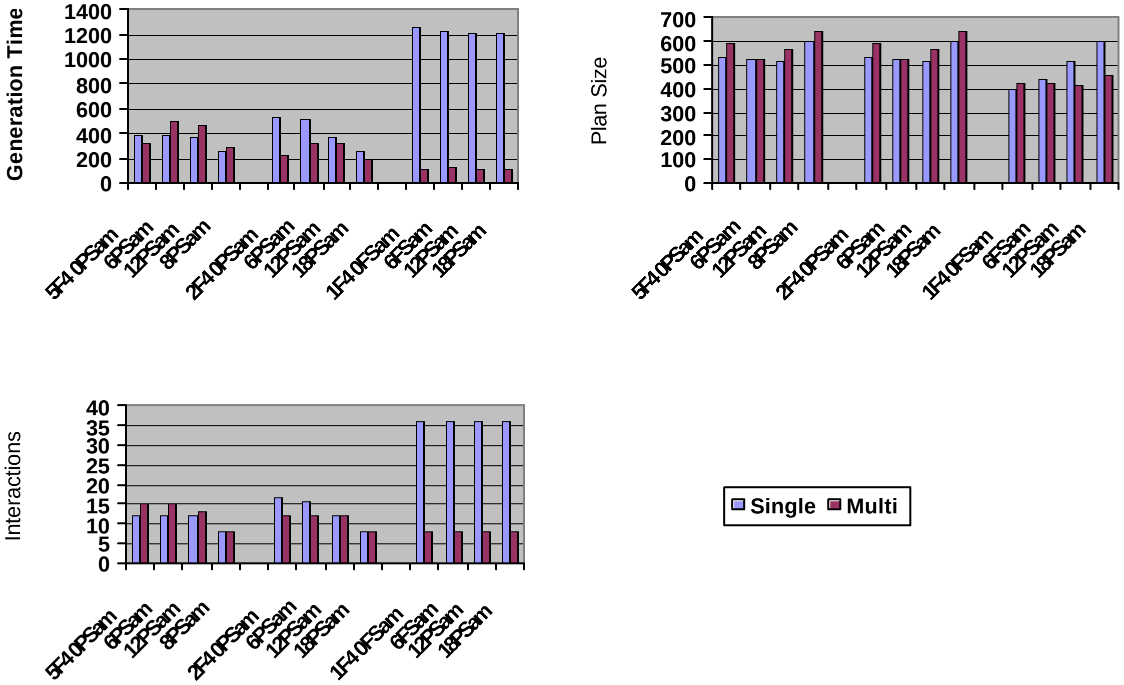

Figure 33. C3 Results: Evaluation of Phantomization Effects

7.3.7. Experiment C4: Sensitivity to the Resource Overutilization Penalty

As noted in Section 7.3.1.3, the parameters P^+ and P^- (see Figure 26) can be adjusted to vary the penalty for resource overutilization and underutilization by operators relative to the established target intensity. In particular, the ratio of P^+ to P^- defines the relative importance accorded to resource underutilization/overutilization.

Experiments C1 through C3 used the values $P^+ =1$ and $P^-= 1$. To assess the sensitivity of the multidimensional approach to these values, we ran the multidimensional strategy with P^+ fixed at 1, but with P^- ranging from 0.5 to 4. Figure 34 displays the results.

For $P^-=4$ (and to some extent, P^-=3), there is a noticeable drop in plan size for the 100% through 50% profiles. For $P^-=.5$, generation times and the number of planner/scheduler interactions are appreciably higher over that same range. Such results are to be expected: when resource overutilization is penalized relative to underutilization (i.e., $P^-/P^+ > 1$), the intensity adaptation process will be more cautious, resulting in a tendency toward smaller plans. In contrast, when resource overutilization is favored relative to underutilization (i.e., $P^-/ P^+ < 1$), the intensity adaptation process will be more aggressive in its strategy selection, possibly resulting in the need for more backtracking due to overly aggressive strategy choices.

We had expected to see more dramatic variation as P^- changed but the adaptive nature of the intensity method appears to compensate for overly aggressive or weak decisions induced by large/small penalty ratios. This robustness makes the intensity adaptation approach strongly insensitive to reasonable values for parameters P^- and P^+.

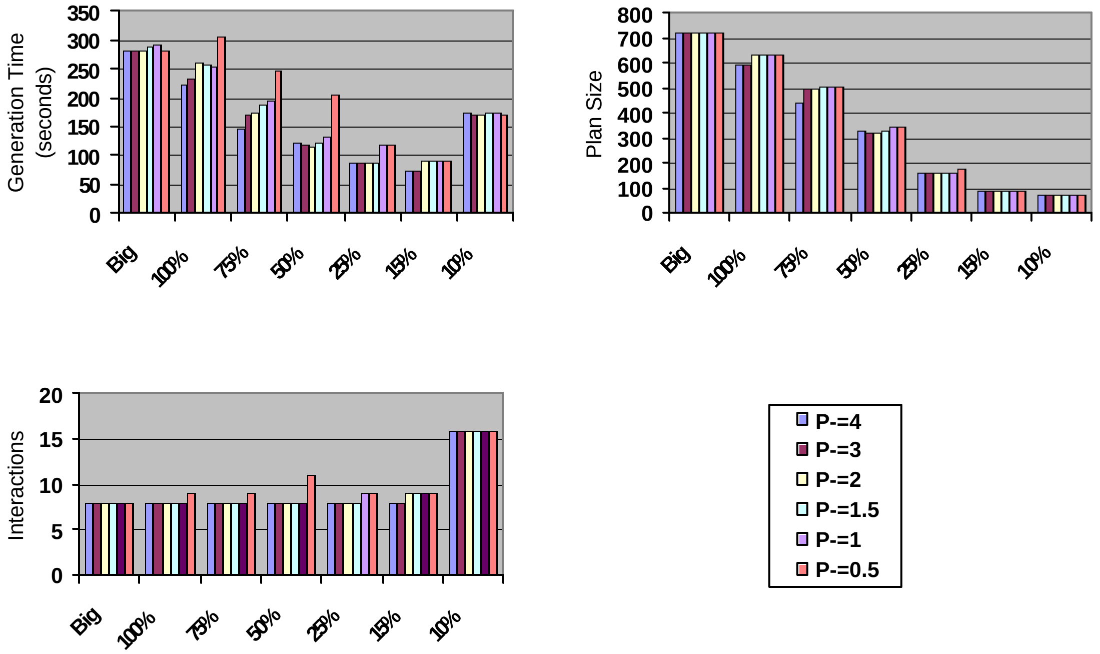

Figure 34. C4 Results: Sensitivity of the Multidimensional Approach to P^-

7.3.8. Experiment C5: Sensitivity to Resource Apportionment Parameter

As discussed in Section 7.3.1.2, capacity determination for each intensity dimension relies on an *allotment weight* that reflects the commander's expectation for relative proportion of resources that will be required for each. Because these allotment weights are an integral part of the capacity calculations, we designed Experiment C5 to evaluate the sensitivity of the intensity adaptation method to these values.

Table 3 shows the range of apportionment distributions considered in this experiment. The ordering of the cases from left to right reflects (roughly) increasing allotment to Precision resources and decreasing allotment to Unguided, Cluster, and Defense resources. Case 3 corresponds to the baseline allotment weights employed in experiments C1 through C4. This baseline was selected as being fairly representative of the distribution of resources for the maximal possible plan (i.e., the plan produced by the Big resource profile). Cases 1 and 2 correspond to situations where Unguided resources have been increased somewhat at the expense of the other three dimensions. Cases 4 through 8 proceed from gradual through to substantial changes away from the baseline allotment.

	Case 1	Case 2	Case 3 (Baseline)	Case 4	Case 5	Case 6	Case 7	Case 8
Unguided	45%	38%	35%	35%	30%	25%	15%	5%
Precision	5%	2%	5%	15%	15%	25%	35%	85%
Cluster	30%	35%	35%	30%	30%	25%	15%	5%
Defense *(SEAD)*	20%	25%	25%	20%	25%	25%	35%	5%

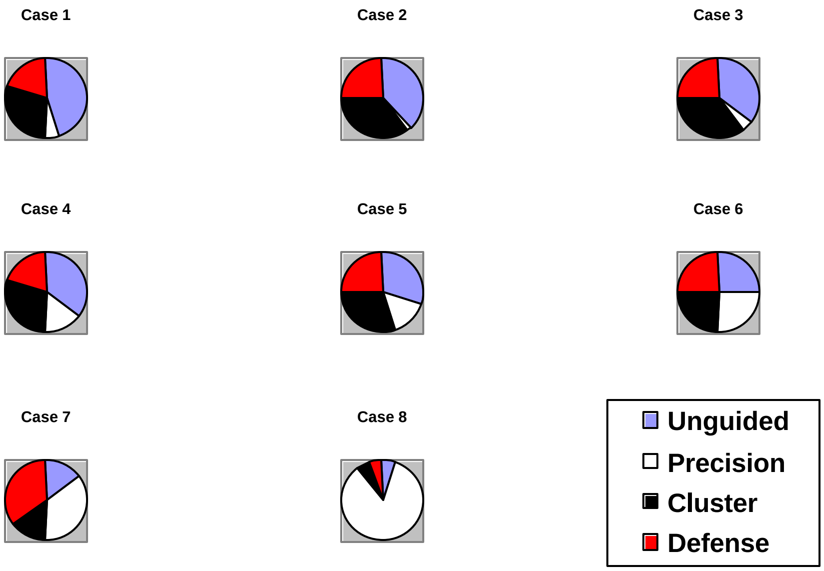

Table 3: Resource Allotment Factors for C5 Sensitivity Testing

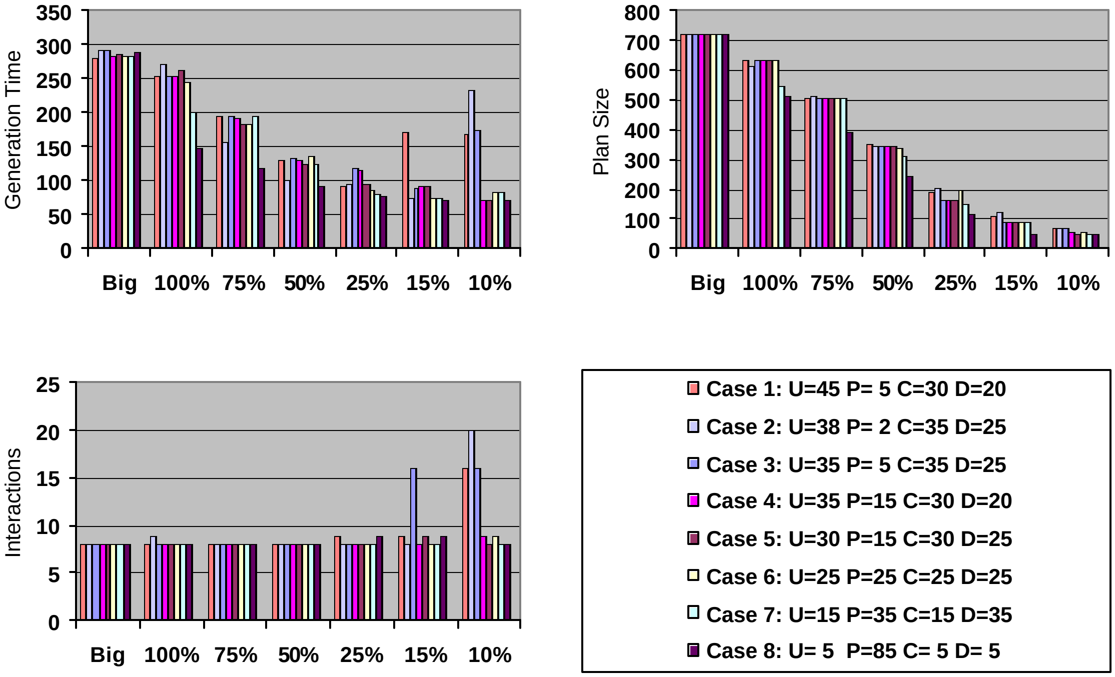

Figure 35. C5 Results: Sensitivity to Resource Apportionment Weights

Figure 35 presents the results for Experiment C5. Overall, they show only minor sensitivity to the allotment weights. As with the testing of sensitivity to P^- in Experiment C4, the explanation for this insensitivity is the adaptive nature of the intensity method. Even the extreme Case 8 behaves reasonably well, despite its significant deviation from the baseline. For this case though, there is a noticeable drop in both plan size and generation time; these are to be expected, as the low allocations for Unguided, Cluster, and Defense would lead to conservative target intensities for those dimensions. Hence, the planner would be underallocating resources for those dimensions, yielding significantly smaller plans than is possible.

One aspect of the graphs does stand out, namely the spikes in generation time and interactions in Case 1 for the 15% profile, and Cases 1 through 3 for the 10% resource profile. These spikes indicate more aggressive strategy than pursued by the other cases for the resource profile. To some extent, this aggressiveness is rewarded with slight increases in plan size. If reduced generation time is considered more important than plan size, these results indicate that different resource allotments may be preferable. Again, though, the overall difference among the cases is relatively small, thus demonstrating the robustness of the intensity method with respect to apportionment weights.

8. Conclusions

Tightly coupled integration of planning and scheduling is essential for effective coordination and agile response in large-scale, dynamic problem domains. Current approaches to automating these functions treat them as separable tasks, which frequently leads to gross inefficiencies, poor coordination, and suboptimal operations.

The JFACC planner-scheduler work summarized by CMU and SRI in this report has developed and evaluated a number of techniques and strategies for increasing the coupling between automated planning and scheduling systems, and achieving the performance benefits of such tighter interleaving. We have focused specifically on the complex domain of air operations planning and scheduling, and in this domain, we have shown how incremental plan synthesis and resource allocation can speed up the problem-solving process and provide greater stability in the plans/schedules generated over time without compromising the overall quality of the solutions generated. At the same time, the techniques we have investigated have broader applicability. They are relevant to any resource and time constrained planning domains where scale and complexity prohibits systematic exploration of all possibilities and practical solutions must rely on incomplete (approximate) search procedures.

To couple planning and scheduling processes, one principal approach explored in this work is grounded in the use of intensity models of expected resource usage. The basic concept is to associate estimates of the expected resource usage with different planning options, and then to use these estimates in conjunction with resource utilization information derived by the scheduler to bias and (as necessary) redirect the plan generation and repair process. This intensity-driven approach to planner/scheduler integration illustrates one way to capitalize on early visibility of resource availability constraints and likewise provides the means to focus replanning effort when execution events (e.g., loss of resource capacity) force changes to the current plan and/or schedule. As shown through the experimental results described in this report, these techniques provide superior performance over a waterfall integration of planning and scheduling technologies, both in terms of reduced generation/repair times and improved plan quality. These benefits are particularly noticeable in highly resource-constrained situations.

One potential problem with the intensity-based method is the reliance on a few key problem-specific parameters. Our experimentation showed, however, that the intensity approach is relatively insensitive to parameter choices within a broad range of reasonable values. This robustness stems directly from the highly adaptive nature of the intensity-based methods, which enables adjustment of strategy on the fly to reflect current and expected resource consumption levels.

In the case of bottom-up reaction to unexpected events such as the loss of resource capacity, a second planner/scheduler integration strategy based on identification and modification of those tasks that have proved difficult to schedule was also investigated and found to work well. This scheduler-driven strategy can be seen as producing a competing bias to that of an intensity-driven approach, and we suspect that in many planning/scheduling circumstances, the optimal response may in fact lie somewhere in the middle. One idea for future investigation is the development of a broader integration protocol, where these competing biases form a basis for negotiation between planner and scheduler.

The examination of incremental replanning and rescheduling problems, both to accommodate new or changed objectives and to respond to unexpected execution events, also gave rise to analysis of the behavior of constituent planning and scheduling processes, validating the desirable incremental properties (e.g., efficiency, solution stability) of our respective solution techniques. In one case, analysis of an initially developed technique for priority-based scheduling found it to be unnecessarily myopic, and this discovery subsequently led to the design and implementation of a much less disruptive and hence much superior solution change procedure.

One aspect of joint planner/scheduler design not considered in this work was integration with the human decision-maker. The SRI planning and CMU scheduling technologies both support flexible, human-in-the-loop generation and repair, through incorporation of various forms of user guidance. Although support for human guidance was not a technical focus for the project, we envision human direction as an integral part of the planning and scheduling process in many (if not most) complex domains. Understanding the mechanics of injecting human decision-makers into automated planning and scheduling processes is clearly another broad direction for future research.

With regard to the experimentation performed in this work, there are also further steps that need to be taken. The experiments documented here have been conducted relative to a specific planning knowledge base and hence make specific assumptions about the structure of the problems to be solved. It would be useful to broaden the experimentation to include other planning domains, and consider more comprehensively the impact of task inter-dependencies and other similar aspects of problem structure. More generally, the reactive experimentation performed in this project has restricted its attention to solution of so-called “point” experiments; given a starting plan/schedule and some external event, how should planning and scheduling processes be engaged to appropriately realign the solution. The obvious next step is to evaluate the integrated planning and scheduling strategies that we have developed in an extended, embedded application context. The appendix provides a proposal for this type of evaluation.

9. References

Becker, M. and S.F. Smith, "Mixed-Initiative Resource Management: The AMC Barrel Allocator", *Proc. 5th Intl. Conf. on AI Planning and Scheduling*, 2000.

Chien, S., G. Rabideau, J. Willis and T. Mann, "Automated Planning and Scheduling of Shuttle Payload Operations", *Artificial Intelligence*, 114(1-2), 1999.

Dean, T., J. Firby and D. Miller, "Hierarchical Planning involving Deadlines, Travel Time, and Resources", *Computational Intelligence*, 4, 1988.

Finin, T., J. Weber, G. Wiederhold, M. Genesereth, R. Fritzson, D. McKay, and J. McGuire, "Specification of the KQML Agent-Communication Language", Technical Report EIT TR92-04, Enterprise Integration Technologies, Palo Alto, CA, 1992.

Jonsson, A., P. Morris, N. Muscettola, K. Rajan and B. Smith, "Planning in Interplanetary Space: Theory and practice", *Proc. 5th Intl. Conf. on AI Planning and Scheduling*, 2000.

McVey, C.B., E.M. Atkins, E.H. Durfee and K.G. Shin, "Development of Iterative Real-Time Scheduler to Planner Feedback", *Proc. 16th Intl. Joint Conf. on AI, 1997.*

Lee, T., "The Air Campaign Planning Knowledge Base", SRI International Technical Report, 1998.

Muscettola, N., S.F. Smith, A. Cesta and D. D'Aloisi, "Coordinating Space Telescope Operations in an Integrated Planning and Scheduling Framework", *IEEE Control Systems*, 12(1), 1992.

Myers, K.L., "CPEF: A Continuous Planning and Execution Framework", *AI Magazine, 20(4)*, 1999.

Myers, K.L., "Strategic Advice for Hierarchical Planners", *Principles of Knowledge Representation and Reasoning: Proc. 5th Intl. Conf.*, Morgan Kaufmann, 1996.

Sadeh, N., D.W. Hildum, T.J. LaLiberty, J. McA'Nulty, D. Kjenstad and A. Tseng. "A Blackboard Architecture for Integrating Process Planning and Production Scheduling, *Concurrent Engineering: Research & Applications*, 6(2), 1998.

Smith, D.E., J. Frank and A.K. Jonsson, "Bridging the Gap Between Planning and Scheduling", *Knowledge Engineering Review*, 15 (1) 2000.

Smith, S.F., O. Lassila and M. Becker, "Configurable Systems for Mixed-Initiative Planning and Scheduling", in *Advanced Planning Technology* (ed. A. Tate), AAAI Press, 1996.

Srivastava, B., S. Kambhampati and B.D. Minh, "Planning the Project Management Way: Efficient Planning by Effective Integration of Causal and Resource Reasoning in RealPlan", *Artificial Intelligence*, to appear, 2001.

Wilkins, D.E., *Practical Planning: Extending the Classical (AI) Planning Paradigm*, Morgan Kaufmann, 1988.

Wilkins, D.E. and K.L. Myers, "A Multiagent Planning Architecture", *Proc. 4th Intl. Conf. on AI Planning Systems*, 1998.

Zhou, Q. and S.F. Smith, “A Priority-Based Pre-emption Algorithm for Incremental Scheduling with Cumulative Resources, Robotics Institute Technical Report, Carnegie Mellon University, forthcoming, 2001.

10. Appendix - Experiment D: Embedded Operations

This section describes a proposed experiment for evaluating the effectiveness of JPS as an embedded controller for the air operations domain. As noted in the introduction to this document, it was decided that the CMU/SRI team should focus on developing and evaluating extended planner/scheduler integration techniques in lieu of performing this experiment. We include the outline of the experiment here for completeness. The description of the experiment conforms to the documentation format provided by the JFACC program.

10.1. Statement of Experimental Objectives

10.1.1. Hypotheses

Hypothesis: Tightly coupled planning and scheduling can provide responsiveness and agility for real-time command and control for air operations.

Level: JFACC Mission Contribution Experiment

Description:

This experiment will involve situating the integrated Planner/Scheduler within an appropriate simulation environment (either the Enterprise Model or SRI's SimFlex simulation framework) in order to evaluate its ability to support realtime command and control for Air Operations. The system will accept information relating to feedback from execution assessment, evaluate impact on current plan/schedule, interleave schedule and plan revision actions as necessary to resolve detected problems and exploit opportunities. The objective is to determine whether the technology can meet the challenges of a realistic operational setting.

10.1.2. Value

Evaluations within an appropriate simulation environment will provide a strong assessment of whether the integrated Planner/Scheduler will be able to cope with the demands of realistic air operations.

10.2. Description of Experimental Setup

10.2.1. Simulation Features

10.2.1.1. Plant

<u>Plan Entities and Dynamics</u>

- Weapons effectiveness models (Scheduler)
- Logistics models (Scheduler)
- Target networks (Planner)
- Centers of gravity (Planner)
- Campaign objectives (Planner)
- Weather (Planner, Scheduler)

Exogenous Data Streams and/or Events

- Pop-up targets
- Intelligence regarding new/changed threats
- Weather
- Redirection of top-level campaign objectives
- Nondeterministic changes in logistics
- Execution results (actual mission execution times, probability of target success)
- Situation assessment (whether low-level objectives are satisfied)

10.2.1.2. Plan or System Identification

Internally, the planner and scheduler maintain the following information about dynamic (projected) state:

- Resource availability and location profiles over time
- Expected takeoff, TOT, refueling, landing times for each mission (i.e., an attack schedule for the current horizon)
- Centers of gravity
- Threats (in terms of source, type, level, expected direction of attack)

10.2.1.3. Control Signals

The control signals will be defined as a set of missions to execute, encoded as activities with the following specified information:

- Designated focus: a target/DMPI, a patrol location, or some other form of air mission objective.
- Assigned wing/base: the base/wing to which the mission has been assigned
- Weaponeering solution: the platform and munitions types to be used, and the numbers of each that are required.
- Execution time frames. Minimally, this will include a TOT window, but this could also specify which launch window to use and other timing information
- Supporting resources and synchronization constraints: In the event that supporting resources are required to carry out the mission (e.g., tanker aircraft to provide refueling), appropriate support missions will also be assigned and synchronization constraints (time, location) will be specified.

The set of missions will be updated dynamically by the planner/scheduler in response to both execution results and changes in tasking, resource availability, and simulated world conditions.

10.2.1.4. State Observation Signals

The following state observations are required:

- Execution results
    - Actual times when the mission launched, when target was hit, etc.
    - Effects – e.g., what probability of destruction was achieved
- Changes in resource availability
- Changes in state information *(Projected – not currently used)*
    - Weather
    - Enemy defenses (locations, types)
- Intelligence
    - Changes in identified threats
- Situation Assessment
    - Determination of success in attaining stated objectives

10.2.2. Variables or Correlated Parameters

Independent Variables:

- Rates of success/failure for missions
- Rates of changes in world state
- Rates of changes in guidance
- Rates of changes in resource availability

10.2.3. Specification of Test Runs

The most appropriate baseline to use for comparison would be comparable runs in which operational personnel develop and adapt the plans and schedules. The cost and difficulty involved in conducting such tests, however, makes them impractical. Furthermore, since there is no current technology that provides comparable capabilities, technological baselines are not possible.

Given the lack of a suitable baseline, our evaluation will focus on measuring the limits of the technology. Limits will be defined in terms of the scope (qualitative) and frequency (quantitative) of changes to which the system can successfully adapt. Success will be measured relative to the achievement of stated objectives as a result of the simulated execution of scheduled plans.

10.3. Pre-Lab Analysis

Similar to the pre-analysis performed prior to Experiment B, we will expand on the results already obtained for different plan and schedule revision strategies and perform preliminary configuration, evaluation and tuning of strategies to determine a composite control strategy for coordinating the planner/scheduler response to (simulated) execution results.